System Identification and Optimization of Fuzzy Relation Matrix Models Based on Semi-Tensor Product

by

Hongli Lyu

A thesis submitted to Lakehead University
in partial fulfillment of the requirements for the
Degree of

Doctor of Philosophy in Electrical and Computer Engineering

LAKEHEAD UNIVERSITY

DEPARTMENT OF ELECTRICAL ENGINEERING

October 15, 2021

Abstract

Generally, in real-world engineering disciplines a dynamical system is nonlinear, having multi-input and multi-output (MIMO) variables, and high-level parameter uncertainties. Although there are many approaches proposed in the literature for system modeling and optimization, it remains a challenging topic to derive the precise mathematical models to characterize complex, dynamic and globally described systems. If training data in a real-world system are available, artificial neural network theories can be applied for system parameter recognition and optimization. The objective of this work is to develop a new fuzzy formulation based on the semi-tensor product (STP) method to construct fuzzy logic models for MIMO systems in a matrix representation. It involves the following processing operations: fuzzy modeling, structure and parameters identification, system optimization, and adaptive control of closed-loop fuzzy systems based on the fuzzy relation matrix (FRM) models and STP algorithms. The related contributions are summarized below:

- The STP operations of logic matrices are proposed for fuzzy logic reasoning by the extension of STP for conventional matrices. And then some properties are developed for the fuzzy STP operations.
- A vector expression of fuzzy variables and matrix representation of fuzzy relations, are proposed for advanced research. Matrix expression of multi-dimensional data is applied to multi-variable fuzzy relations and fuzzy rules.
- Two modeling methods, direct modeling and indirect identification are proposed to identify the FRM models for MIMO fuzzy systems.
- A universal approximation with the approximation accuracy is proposed for MIMO fuzzy models based on FRMs and the fuzzy logic STP algorithm.

- A neural-fuzzy STP network is developed to train the parameters in MIMO fuzzy systems based on the FRM and fuzzy logic STP algorithm, and a hybrid optimization method is adopted to train parameters for FRM models.
- Based on the FRM model and fuzzy logic STP algorithm, an indirect adaptive FRM control law is suggested to improve the performance of FRM control systems.

The effectiveness of the proposed modeling, optimization, and adaptive control design techniques in the multi-variable FRMs and STP algorithms platform is validated by simulation tests in the Matlab environment.

Key Words: System identification; Semi-tensor product (STP); Fuzzy relation matrix (FRM); MIMO systems; Matrix representation; Adaptive control.

Acknowledgments

I have taken efforts in this thesis in the past four years. However, it would not been possible without the kind support and help of many individuals in Lakehead University.

First and foremost, I would like to thank Dr. Wilson Wang for giving me the opportunity to pursue a Ph.D. under his supervision at Lakehead University. I am tremendously grateful for his enthusiastic supervision and continuous support, for countless inspiring discussions, and for giving me the freedom to pursue a variety of research directions in the past four years. I heartily acknowledge his constant encouragement to explore possible funding routes for the continuation of my research studies.

I would like to express my great gratitude to Dr. Xiaoping Liu as my co-supervisor and thank him for giving me the extraordinary opportunity to be a Ph.D. student at Lakehead University. His knowledge, encouragement, and supportive mentorship help me in my whole doctoral study. I appreciate Dr. Liu for teaching me how to do rigorous research carefully and make progress in my past studies. I also thank him for his funding support and help in the past four years.

I appreciate deeply my external examiner, Professor William Melek in the Department of Mechanical and Mechatronics Engineering at the University of Waterloo, for examining my thesis carefully and attending my thesis defense.

I appreciate sincerely my Ph.D. committee members Dr. Kefu Liu and Dr. Abdelhamid Tayebi, for their professionally instructive guidance and patient assistance, which help me to graduate so smoothly.

Great thanks also go to my Coordinator, Dr. Abdelhamid Tayebi, for his patient guidance and always-cheerful conversations in the past three years. I also thank him for excellent lectures about control theories, which are very helpful to my research. I also appreciate my current Coordinator, Dr. Salama Ikki, for his work on my thesis examination and supporting my graduation.

Sincere appreciations to other professors at Lakehaed University, Dr. Rachid Benlamri, Dr. Shan Du, Dr. Meilan Liu, Dr. Ruizhong Wei, Dr. Qiang Wei, Dr. Hui Zhang, Dr. Zihua Jiang, Dr. Will Zhao, Dr. Shikui Wu, Dr. Baoqiang Liao, and Dr. Wensheng Qin, for all of your warm guidance and favors.

Being a part of Lab CB1034, the Lab for Intelligent Mechatronics Systems, was a wonderful and truly enriching experience. I have fond memories of the "early days" in our lab where I shared an office with Peter, Shazali, Zihao, Manzar, Yichao, Jital, Ahwiadi, et al.

Special thanks to my Aunt Maureen and Uncles Dennis, Aunt Lois and Uncle Rudy, my best Canadian friends in Thunder Bay. They gave me a very warm parents-like happiness and comfortable dependence since I reached to Thunder Bay in 2017.

Finally, I would like to thank my parents for their unconditional continuous encouragement, support, and love, which has made this extensive research task a pleasant journey. To my younger brother, Xiaosheng, thank you for understanding me and also taking care of our parents to make me peaceful in Thunder Bay. I appreciate my daughter, Candy, for her significant companying every day. I would not imagine where I am today without having her in my life.

So many thanks also go to everyone from both near and far for enriching the past four years in my life, providing support and welcome distraction, and simply for being in my life. I could not wish for a better person to share my journey with all of you.

Sincerely,

Hongli (Mary) Lyu

October, 2021

Table of Contents

Table of Figures

Table of Acronyms

AI	artificial intelligence
ANFIS	adaptive neural fuzzy inference systems
ANN	artificial neural networks
FRM	fuzzy relation matrix
FLC	fuzzy logic control
GA	genetic algorithm
GD	gradient descent
LSE	least squares estimator
MF	membership function
MIMO	multi-input and multi-output
MISO	multi-input and single-output
NF	neural-fuzzy
NF-STP	neural-fuzzy semi-tensor product
PID	proportional-integral-derivative
RLM	recursive Levenberg-Marquaedt
STP	semi-tensor product
SISO	single-input and single-output
TISO	two-input and single-output
TITO	two-input and two-output
TSK	Takagi-Sugeno-Kang

Chapter 1 Introduction and Literature Review

1.1 Research Motivation

With the wide application of computing devices, most real-world systems have become more and more complex. Correspondingly, it becomes more challenging to precisely describe their dynamic characteristics especially for those with multi-variables, nonlinearity, uncertain parameters, and unknown disturbance, based on exact mathematical models or classical mathematical formulation and analysis. Although a great deal of effort has been undertaken in the past decades for theoretical study of linear systems, most of the traditional and modern system theories are constructed on the basis of deterministic expressions such as using polynomial functions, differential equations, state matrices, probability, and statistics [1]. As a result, it still remains a challenging research topic to construct precise nonlinear mathematical models for system identification, analysis and synthesis [2].

In general, a good engineering system model should make use of available data and information effectively. In application, some important information comes from expertise knowledge and reasoning, sensor measurements, data analysis, as well as physical laws. Therefore, it is desirable to combine these types of information into system modeling and design. Hence, the critical question is how to efficiently identify suitable mathematical models based on the sampling input-output data for different applications [3, 4].

1.2 Brief Background

Generally, the analysis for linear and nonlinear multi-input and multi-output (MIMO) systems concentrates on the use of conventional system identification methods, such as frequency-domain analysis, time-domain analysis, transfer function, state-space expression, differential equations, limit cycle theory, phase plane methods, describing functions, and Lyapunov stability theories, etc.[5, 6]. The most common method could be linearization in a limited discourse so that the linear control system theories can be used for locally or piecewise linearized nonlinear models [7]. However, most of these analysis techniques are on the basis of exact mathematical models, where precise models are difficult to derive especially for complex multi-variable systems with coupled multi-output variables. On the other hand, some expertise and/or knowledge can provide valuable insight to system identification, reasoning, and operation, even though it could be difficult to obtain accurate mathematical models. Under these linguistic knowledge representations, fuzzy systems could provide a promising alternative for system modeling and performance control when the accurate mathematical model of a process is difficult to derive [8, 9].

The fuzzy logic theory is an effective tool to formulate human knowledge in a systematic manner by the use of fuzzy IF-THEN rules. As an important element of artificial intelligence (AI), fuzzy logic theory has been making accelerating progress in recent decades especially in fuzzy control applications to consumer products and industrial control [10]. However, it is difficult to incorporate a fuzzy model with other conventional mathematical expression for advanced modeling and analysis, since fuzzy rules and reasoning processes are realized by linguistic and logic operations [11].

The semi-tensor product (STP) of matrices was first proposed by Cheng in 2011 [12], which is a useful mathematical tool to express binary logic and multi-valued. The STP of matrices can realize the multiplication operation between two arbitrary matrices without following the conventional necessary conditions for matrix product such as the dimension of the column for the first matrix has to be equal to the dimension of the row for the second matrix. Therefore, it can provide an efficient tool to tackle many problems related to matrix operations. As a generalization of matrix product operation, the STP not only can keep all the properties of the conventional matrix product operations, but also can have its unique properties such as pseudo-communicative law [13]. Hence, STP has been applied to many research fields related to matrix operations [14], including the analysis and control of multi-valued logic and mix-valued logic inference systems through matrix representation of logical variables [15, 16]. It also has the potential to be used with some soft-computing paradigms such as artificial neural networks (ANNs) and genetic algorithms, to facilitate the design and implementation of intelligent systems [17].

1.3 Literature Review

Most real-world systems are nonlinear and continuous in nature, which do not obey the superposition principle [18, 19]. Compared with linear systems, there are very limited mathematical tools that can be used to nonlinear MIMO system analysis, except differential equations, limit cycle theory, Lyapunov stability theorems, and describing functions etc. [20-22]. If solutions near an equilibrium position are of interest only, a nonlinear system could be approximately linearized over a local region [23, 24]. Typically, the behavior of a nonlinear system is often described mathematically by nonlinear equations with unknown and uncertain parameters. However, as it is usually difficult to solve nonlinear equations precisely, nonlinear systems are commonly approximated by linear equations. On the other hand, because some phenomena like chaos and singularities could be ignored by linearization, some aspects of dynamic behaviors of a nonlinear system could become counterintuitive, unpredictable, even chaotic or random. As a result, it is critical to develop appropriate mathematical modeling methods for nonlinear system analysis and control [25].

1.3.1 Introduction to fuzzy logic inference theory and applications

The fuzzy logic theory was initiated by Zadeh in 1965 [26], by using fuzzy sets, fuzzy logic, and fuzzy IF-THEN rules to formulate human knowledge [27-29]. Generally, based on the type of the consequent reasoning in fuzzy rules, fuzzy systems can be categorized into Mamdani models, Takagi-Sugeno-Kang (TSK) models, and dynamical fuzzy models [30-32]. Hence, the rules-based fuzzy system can provide many promising results [33], especially, fuzzy control theories was applied successfully in real world since 1970s [34-37]. For example, Lee proposed a robust adaptive fuzzy controller by backstepping for a class of MIMO nonlinear systems [38]. Li *et al.* have investigated the latticized linear programming subjected to fuzzy relation inequality constraints with the max-min composition [39]. Feng *et al.* have proposed a fuzzy dynamic model and the related synthesis theory in [40]. Li *et al.* have represented a class of robust adaptive-fuzzy-tracking control for nonlinear multi-variable systems by a fuzzy approximation approach [41].

Additionally, fuzzy controllers have found many successful applications in real-world systems, but generally using simplified multi-input and single-output (MISO) fuzzy models. Besides, it is difficult to use MIMO fuzzy models in real-world systems because the fuzzy output variables are usually coupled and dependent on each other. Accordingly, the fuzzy rules may not be accurate enough to represent the relationship between each input and each output for MIMO fuzzy controllers [42, 43]. To date, the fundamental challenge in this research area is the lack of efficient mathematical modeling strategy to deal with extensive databases [44, 45]. Thus, a new analytical strategy is needed to improve the performance of conventional fuzzy models.

On the other hand, fuzzy logic has been applied in many fields such as system control and decision, signal processing, communications, integrated circuit manufacturing, and expert systems [46]. However, most of its applications are related to system control [47] including fuzzy washing machines, digital image stabilizers, fuzzy systems in cars, fuzzy control of subway train, etc. [48]. Moreover, if the training data can be used to improve the modeling accuracy of fuzzy systems, the application of fuzzy systems will be expanded significantly. With the combination with other processing strategies, especially with soft-computing tools, the performance of fuzzy systems can be significantly improved for different applications [49, 50].

1.3.2 Modeling of MIMO fuzzy systems

In general, for a general multi-variable system, system identification is the first step to model a nonlinear system, which is to obtain information about the kernels or transfer functions of the unknown system from input-output experiments [51]. It is usually composed of two sequential procedures: structure identification and parameter identification. In system identification, the dynamic system of interest can be characterized by mechanical, physical, chemical, or biological objects, and then processed according to some information transformation algorithms [52]. The models can be further improved through the comparative analysis between the theoretical and experimental results [53].

Fuzzy logic inference models have been widely used in many industrial and commercial plants applications [54, 55], since it was firstly applied by Mandani to control steam engine locomotive system in 1974 [30]. Generally, a fuzzy model is built based on fuzzy IF-THEN rules that could be inferred from linguistic knowledge or expertise from operators and experts. Then, the fuzzy model can be realized by structure identification and parameters optimization through algorithms such as fuzzy approximation, fuzzy clustering, evolutionary computing, and immune modeling, etc. [56-58]. However, with the rapid development of the internet and computer technology including microprocessors, the information processing abilities are also increasing [59], so that, there are no clear advancements in modeling and optimization in traditional fuzzy systems [60-63]. One of the reasons may be related to the uncertainty in fuzzy linguistic description. In addition, it is difficult to model and optimize fuzzy systems by directly using the traditional computing strategies based on vectors and matrices in programming, digital strategies, discrete mathematics, multi-value logic, etc. [64].

1.3.3 Universal approximation of fuzzy models

Since most systems are nonlinear, MIMO, and subjected to unpredictable disturbance, and variable operating conditions, it is challenging to derive precise mathematical models. Although the universal approximation capability of fuzzy logic systems is studied for nonlinear system identification [65, 66], these fuzzy approximators are composed of conventional fuzzy rules and fuzzy reasoning process. Therefore, there exists some limitation for them to estimate any nonlinear functions, especially the MIMO systems. But, many techniques have been used to improve the universal approximation for the fuzzy rule-based models. A rough set method has been introduced in [67] to approximate preference relations using fuzzy inference with multi-attribute dominance. The mean approximation of regular fuzzy neural networks is investigated in [68] for Choquet integral. The concept of intuitionistic fuzzy S-approximation spaces is suggested in [69] to process the uncertainty in fuzzy decision-making. The aforementioned methods can provide the fundamental tool to apply approximation theory to fuzzy systems, however, most of the current results focus on the fuzzy model based on fuzzy rule sets and conventional fuzzy logic reasoning process.

On the other hand, the universal approximation theory in [32] can be applied in the analysis and synthesis of some types of nonlinear systems [70, 71], such as fuzzy modeling with second-order approximation [72], adaptive fuzzy neural network approximation [73], reinforcement learning [74], and fuzzy adaptive control of nonlinear systems [75, 76]. For example, an observer-based adaptive fuzzy control is proposed in [77] to approximate the unknown nonlinear functions for a class of MIMO non-strict feedback nonlinear systems; a backstepping state feedback controller is developed in [78] for nonlinear MIMO time-delay systems based on fuzzy approximation. In these works, fuzzy logic is used to approximate unknown nonlinear functions, but a series of MISO fuzzy rules-based systems are designed as approximators in the respective MIMO nonlinear systems. The universal approximation of an MIMO nonlinear function is seldom to be considered directly due to lacking available mathematical tools.

1.3.4 Optimization of fuzzy models

The modeling accuracy can be improved by optimization of system parameters, subjected to some physical constraints and performance requirements. In general, the optimization operation can be classified as static optimization and dynamic optimization [79]. Static optimization is relevant to a plant with steady-state conditions, or the system variables are not changing with time. The plant can be described by algebraic equations by the use of ordinary calculus, Lagrange multipliers, linear and nonlinear programming, etc. [52]. On the other hand, dynamic optimization is corresponding to plants with time-varying properties by the use of search techniques such as dynamic programming, calculus of variations, and Pontryagin principle of optimality, etc. [80].

Generally, fuzzy optimization process is usually composed of two operations: fuzzy decision and parameter identification [81]. The fuzzy decision [82] can be undertaken by the use of fuzzy linear programming models, fuzzy nonlinear programming models, fuzzy dynamical programming, feasibility linear programming models, etc. [83]. Several algorithms are proposed in recent years for structure and parameter optimization of fuzzy logic systems, such as fuzzy sorting, fuzzy sets operation, sensitivity analysis, and dual theory [84]. On the other hand, in order to improve the rigorous fuzzy models in conventional rules-based fuzzy systems, an appropriate optimization algorithm is selected to update system parameters based on the proper objective functions or design requirements. Some conventional optimal strategies include Least squares estimator (LSE) and gradient descent algorithms, etc. [85], while more advanced nature-inspired optimization algorithms consist of ant colony optimization [86], particle swarm optimization [87], and simulated annealing, etc. [88-90].

1.3.5 Fuzzy control of MIMO systems

A nonlinear dynamical system is regarded as a challenging endeavor in both theory and applications. Many physical processes are represented by nonlinear models in applications such as process control, biomedical engineering, robotics, aircraft, and spacecraft control, etc. [91]. More advanced control laws are needed to meet stringent design specifications of nonlinear control systems [92]. Most systems are inherently nonlinear, even though nonlinear systems could be simplified as linear under certain operating and application conditions [93]. Hence, compared with the wide variety of techniques available for linear system analysis, the tools for the analysis and design of nonlinear systems are very limited [94].

Conventional fuzzy models based on fuzzy rules and fuzzy logic reasoning are generally used as nonlinear controllers. There are three main types of fuzzy control systems commonly used. The first type is related to classical fuzzy controller, including self-tuning, self-supervised, self-adaptive fuzzy controllers [95-97]. The second type is related to the fuzzy PID controller, where the proportional, integral, and derivative tuning parameters can be adjusted by proper fuzzy operations [98, 99]. The third type is related to synergistic system control, such as fuzzy expert control, neural fuzzy control, and adaptive neural fuzzy inference systems (ANFIS) control [100-102]. In general, the output variables of a fuzzy controller are associated with control laws for the plant, and the input variables are related to the controlled system. The basic fuzzy operations include fuzzification, fuzzy inference, fuzzy rules set, and defuzzification, etc. [103]. To simplify programming, in the current research results, an MIMO rules-based fuzzy system is usually decomposed into several two-input and single-output (TISO) and/or MISO fuzzy models [104].

1.3.6 Semi-tensor product of matrices as a mathematical tool

The STP of matrices is a generalization of the conventional matrix product, which has a potential to be used in mathematical modeling and system identification applications [12].

In general, the traditional matrix product operation must satisfy some necessary conditions. However, the STP can soften these limitations, which has been effectively used in many applications such as converting a logical relationship into a standard discrete-time dynamic algebra operation [13, 14, 105], logic analysis [106, 107], and games [108, 109]. Some research results have been undertaken in Boolean networks based on STP [110-113], the Lyapunov stability and construction of Lyapunov functions [114], and STP-based stability analysis of Boolean control networks [115, 116]. The STP can also represent fuzzy logic operators and fuzzy reasoning processes with matrices of MIMO fuzzy systems [117].

In system science and engineering, the STP method has two main possible applications. The first is related to modeling and control of nonlinear systems such as Morgan's problem [118], stability region [119], feedback linearization [120], and symmetry of control systems [121]. The other area is related to the analysis and control of logical dynamic systems represented by algebraic state-space models [122-124]. On the other hand, the STP method has been applied to some real applications including gene regulation [125], power system [126], wireless communication [127], smart grid [128], finite automata [129], information security [130], vehicle control [131], indoor thermal comfort [132], fault detection of circuits [133], spacecraft [134], epidemic vaccination [135], and mobile robot [136].

In particular, the STP method can be extended to multi-valued logic and mix-valued logic operation, logic mapping, and logic functions [137]. For example, it can express the logical variables in a matrix form and realize the logic reasoning process. The author's research teams have undertaken a series of innovative research works to extend the STP to fuzzy systems [113, 117, 138-140]. However, these works have considered the discrete-time fuzzy systems only, and the initial matrix model is constructed without considering the different membership functions (MFs), fuzzy reasoning operations, parameter identification, etc. Thus these works could be considered as the primary efforts to transform the matrix description to fuzzy logic modeling.

1.4 Research Challenges

From the above literature review, although the traditional fuzzy logic model based on a set of fuzzy rules has many merits in system modeling and control, it has some limitations in applications. Since the STP method can extend the conventional matrix product operation and realized the matrix expression of multi-valued logic to a more general domain, it has a potential to extend the functionality of traditional fuzzy logic inference systems. Hence, the following challenging topics are proposed.

(1) Both the conventional fuzzy relation matrix and composition operation can be realized for two fuzzy sets or fuzzy variables, but it lacks mathematical expression for multiple fuzzy sets, fuzzy variables or fuzzy relationships. The MIMO fuzzy logic rules-based models cannot be expressed in a matrix form. Moreover, the fuzzy reasoning operation cannot be realized by conventional matrix operators.

(2) STP of matrices can be used to express the logic operations such as AND and OR in the logic reasoning process, so that STP can be used to matrix express the multi-valued logic reasoning process instead of multi-valued mappings and functions. As fuzzy logic is one type of logic algebra with fuzzy relations and fuzzy inference, it is possible to be expressed by matrices and STP algorithms for fuzzy logic inference systems.

(3) The fuzzy models are mainly knowledge-based linguistic description. The sampling database is rarely used to identify the structure and parameters in the traditional fuzzy modeling. If the matrix expression can be realized for the conventional fuzzy logic systems, then, the modeling of fuzzy systems can be constructed based on the input-output sampling database.

(4) Universal approximation of traditional fuzzy logic models mostly focuses on analyzing the TISO and MISO functions rather than general MIMO nonlinear functions, due to lacking the necessary mathematical tool to express the multi-variable fuzzy logic systems. When the general multi-variable fuzzy logic systems can be expressed by the fuzzy relation matrices, the corresponding approximation can also be analyzed based on the MIMO FRM models.

(5) Fuzzy logic systems lack the capability of online self-learning. And generally, fuzzy parameters cannot be updated by training data. When the matrix expression of fuzzy logic reasoning process can be realized as the algebraic model and digital logic

operation, it will be easier for the fuzzy logic inference systems to train fuzzy parameters by some optimization algorithms.

(6) It is difficult to use the classical models-based system theories to improve fuzzy models because there is no precise mathematical model to represent multi-variable fuzzy relations and fuzzy logic reasoning process. Once the matrix expression can be realized for MIMO fuzzy models, the conventional theory and applications can be implemented by FRM models. Then, the relevant theoretical expression can be constructed for conventional fuzzy logic systems based on the FRM models and STP operations.

1.5 Objective Strategies

To tackle the limitations of existing MIMO fuzzy system theories, a new fuzzy formulation platform will be proposed based on the STP algorithm and matrix expression of multi-dimensional data. The goal of this research work is to apply the STP to realize algebraic representation for conventional MIMO fuzzy logic inference systems. It will systematically investigate the issues in system identification, approximation analysis, parameter training, and controller design for MIMO fuzzy systems. The specific research objectives are listed below:

1) Fuzzy STP operations are proposed by extension of STP from conventional matrices to multi-valued logic matrices. The vector and matrix expression of multi-dimensional data is developed for the matrix expression of multi-variable fuzzy relation.

2) The theoretical matrix formulation is proposed for multi-variable fuzzy systems. The general definition of fuzzy relation matrix (FRM) is formulated based on vector expression of fuzzy variables and fuzzy sets. Then, the matrix expression is developed for the MIMO fuzzy rules-based systems based on STP of logic matrices.

3) The FRM models are identified by a direct modeling and an indirect identification strategy using sampling input-output training data. Then, it is demonstrated that a general fuzzy system with an FRM model is a nonlinear mapping from input to output variables, which can be represented by the product of two matrices.

4) A universal approximation is proposed for a nonlinear multi-variable function via FRM models and fuzzy logic STP algorithms. This approach can be used to design arbitrary fuzzy systems with FRM models and to approximate nonlinear functions.

5) FRM model parameters will be optimized based on a novel neural-fuzzy STP network and fuzzy logic STP operations, where the recursive least squares estimator and the recursive Levenberg-Marquaedt algorithms are used to optimize the linear and nonlinear parameters, respectively.

6) A closed-loop FRM control system is constructed by using fuzzy logic STP and an indirect adaptive fuzzy controller is constructed on the basis of FRM models for a *n*-th order nonlinear system. An adaptation law is proposed to optimize the unknown parameters. The effectiveness of the proposed FRM control design techniques is verified by simulation tests.

1.6 Outline of the Thesis

Based on the challenging topics in Section 1.4 and the objective proposed in Section 1.5, the overall structure of this dissertation is described in Fig. 1.1.

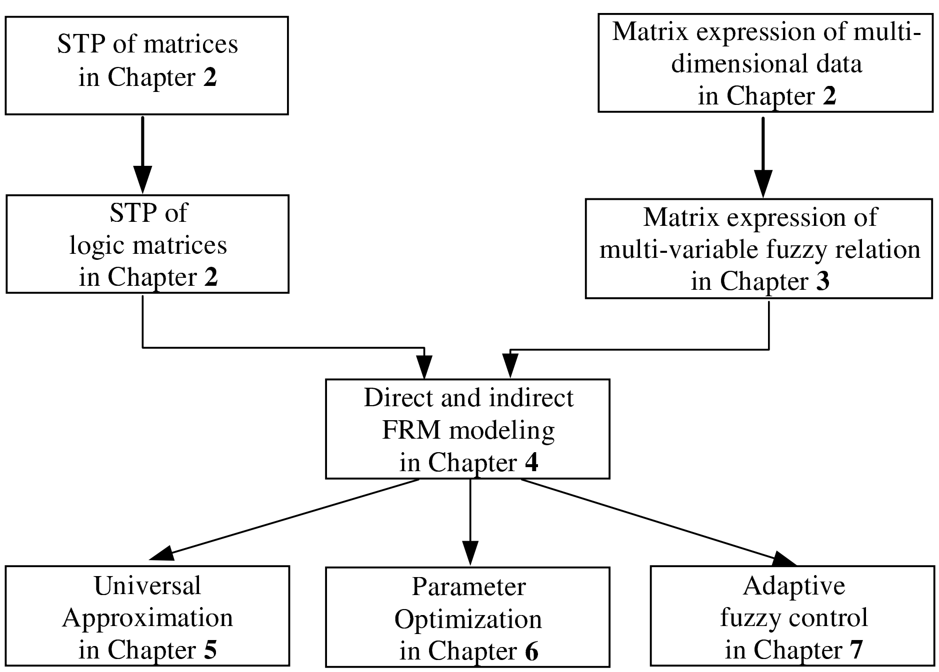


Fig.1.1 The structure of the whole thesis.

The main contents of this dissertation are organized as follows:

In Chapter 2, the STP of logic matrices is proposed, in addition with the related preliminaries about the STP of matrices.

In Chapter 3, vector expression is defined for fuzzy sets and fuzzy variables; the matrix expression of multi-variable fuzzy relations is proposed to formulate a novel fuzzy model.

Chapter 4 discusses the proposed methods for direct modeling and indirect identification of the fuzzy matrix expression for MIMO fuzzy systems.

In Chapter 5, a universal approximation approach of MIMO fuzzy systems is proposed based on the FRM and STP methods.

Chapter 6 discusses parameter training and optimization by using a new neural-fuzzy network based on FRM and fuzzy STP algorithms.

In Chapter 7, the proposed FRM model is applied to design adaptive FRM controller for fuzzy systems with unknown parameters.

Chapter 8 summarizes the main conclusions and contributions in this thesis and some ideas in the future work.

Chapter 2 Semi-Tensor Product of Matrices

The semi-tensor product (STP) of matrices is a fundamental mathematical tool for the whole work in this thesis and the proposed fuzzy relation matrix (FRM) depends on STP algorithms to formulate the novel theoretical architecture for fuzzy logic inference. In this chapter, the relevant mathematical knowledge about the STP algorithm is introduced in order to describe the related contributions in the following chapters.

Firstly, some related notational definitions and preliminaries will be given for STP of matrices in Section 2.1, followed by the multi-dimensional data in Section 2.2. Section 2.3 will introduce the basic definitions and properties of STP of matrices, and then the STP operation of logic matrices will be proposed.

2.1 Introduction of the Related Notations

Firstly, some necessary notations are introduced, which will be used throughout the thesis:

- $D=\{0,\ 1\}$: "1" means “true or *T*” and "0" means “false or *F*”. A logical variable x takes value from D, which is expressed as $x\in D$.
- $D_k=\left\{0,\frac{1}{k-1},\frac{2}{k-1},\cdots,\frac{k-2}{k-1},1\right\},\ k\geq 2$: A set of k-index logic scalars, where k is a positive integer. A logical variable x takes value from D_k, which is expressed as $x\in D_k$.
- $D_k^{m\times n}$: A set of $m\times n$ k-index logic matrices with their entries in D_k, which are called the k-valued logic matrices.
- $B_{m\times n}=D_2^{m\times n}$: If $A\in B_{m\times n}$, A is called a Boolean matrix.

- Z^+ : A set of positive integers.
- $\Re^n$: A set of n-order real column vectors.
- $\Re^{m\times n}$: A set of $m\times n$ real matrices.
- $lcm\{a,b\}$: The least common multiple of a and b, where $a,b\in Z^+$.
- $A\succ_t B$: The dimensional relation is $n=tp$, $A=(a_{ij})\in\Re^{m\times n}$ and $B=(b_{ij})\in\Re^{p\times q}$.
- $Row_i(A)$: The i-th row of the matrix A, $A\in\Re^{m\times n}$.
- $Col_j(A)$: The j-th column of the matrix A, $A\in\Re^{m\times n}$.
- T-norm [141]: A binary algebraic function T: $[0,\ 1]\times[0,\ 1]\to[0,\ 1]$. It satisfies the following properties:
  - Commutativity: $T(a,b)=T(b,a)$
  - Monotonicity: $T(a,b)\le T(c,d)$ if $a\le c$ and $b\le d$
  - Associativity: $T(a,T(b,c))=T(T(a,b),c)$
  - The number 1 acts as the identity element $T(a,1)=a$
  - The T-norm operators commonly used are as follows:
    a. $\{x\wedge y=min\{x,\ y\}\}$
    b. $\{x\cdot y=xy\}$
    c. $\{x\oplus y=max\{0,x+y-1\}\}$
- S-norm (T-conorm) [141]: A binary algebraic function S: $[0,\ 1]\times[0,\ 1]\to[0,\ 1]$. It satisfies the following properties:
  - Commutativity: $S(a,b)=S(b,a)$
  - Monotonicity: $S(a,b)\le S(c,d)$ if $a\le c$ and $b\le d$
  - Associativity: $S(a,S(b,c))=S(S(a,b),c)$
  - Identity element $S(a,0)=a$
  - The S-norm operators commonly used are as follows:
    a. $\{x\vee y=max\{x,\ y\}\}$

b. $\{x \dotplus y = x + y - xy\}$

c. $\{x \otimes y = min\{1, x + y\}\}$

- De Morgan's law [141]: is generalized to describe the relationship between *T*-norm and *S*-norm:

$$S(a,b) = 1 - T((1-a),(1-b)).$$

Definition 2.1: Consider $D_k = \left\{0,\ \frac{1}{k-1},\ \frac{2}{k-1}\ ,\cdots,\ \frac{k-2}{k-1},\ 1\right\}$, where $k \geq 2$ is an integer. Given two logical variables $\alpha, \beta \in D_k$, the operations are as follows [12]:

- Boolean Addition:

$$\alpha +_B \beta = \alpha \vee \beta,\ \alpha, \beta \in D_k. \tag{2.1}$$

- Boolean Product:

$$\alpha \times_B \beta = \alpha \wedge \beta,\ \ \alpha, \beta \in D_k. \tag{2.2}$$

where "$\vee$" and "$\wedge$" can be chosen as an *S*-norm operator (e.g., max) and a *T*-norm operator (e.g., min), respectively. In general, the following notations can also be used for multiple logical variables.

- Boolean Multi-Addition:

$$\sum_{i=1}^{n}{}_B\, \alpha_i := (+_B)_{i=1}^{n}\, \alpha_i = \alpha_1 +_B \alpha_2 +_B \cdots +_B \alpha_n. \tag{2.3}$$

- Boolean Multi-Product:

$$\prod_{i=1}^{n}{}_B\, \alpha_i := (\times_B)_{i=1}^{n}\, \alpha_i = \alpha_1 \times_B \alpha_2 \times_B \cdots \times_B \alpha_n. \tag{2.4}$$

where $\alpha_i \in D_k$, $i = 1, 2, \cdots, n$.

Definition 2.2 [12]: Assume $A = (a_{ij}) \in \Re^{m\times n}$. A is a multi-valued or *k*-valued matrix, with entries $a_{ij} \in D_k,\ 2 \leq k \leq \infty$.

When $k = 2$, A is a binary-valued matrix, usually a Boolean matrix.

When $k=\infty$, $D_\infty := [0,\ 1]$, A is a fuzzy matrix.

$D_k^{m\times n}$ denotes a set of k-valued $m\times n$ matrices. When $m=1$, A is a row k-valued vector. D_k^n denotes a set of the n-dimensional k-valued vector. When $n=1$, A is a column k-valued vector. D_k^m denotes a set of the m-dimensional k-valued vector.

The two basic logic operations can be extended to the following definitions for matrices with their entries over D_k, if the matrix can be expressed into a fundamental k-valued logic.

Definition 2.3 [12]:

- If $A=(a_{ij})\in D_k^{m\times n}$ and $B=(b_{ij})\in D_k^{n\times p}$, the operator of logical addition for logic matrices will be:

$$A\oplus_B B = C = (c_{ij})\in D_k^{m\times p}, \tag{2.5}$$

where $c_{ij}=\sum_{q=1}^{n}{}_B\,(a_{iq}+_B b_{qj}),\ i=1,\cdots,m;\ j=1,\cdots,p$.

- If $A=(a_{ij})\in D_k^{m\times n}$ and $B=(b_{ij})\in D_k^{n\times p}$, the operator of logical product for logic matrices will be:

$$A\otimes_B B = D = (d_{ij})\in D_k^{m\times p}, \tag{2.6}$$

where $d_{ij}=\sum_{q=1}^{n}{}_B\,(a_{iq}\times_B b_{qj}),\ i=1,\cdots,m;\ \ j=1,\cdots,p$.

- If $\alpha\in D_k$ and $A=(a_{ij})\in D_k^{m\times n}$, then,

$$\alpha\otimes_B A = A\otimes_B \alpha = (\alpha\wedge a_{ij})\in D_k^{m\times n}. \tag{2.7}$$

2.2 Multi-Dimensional Data

2.2.1 The related Definitions

Generally, there are many types of datasets in real-world systems. To facilitate analysis, the following definition is given to describe multi-dimensional data.

Definition 2.4 (**Multi-dimensional data**) [137]: Suppose a multi-dimensional data is related to k factors $i_1, i_2, \cdots, i_k$, and each factor i_j has n_j levels from 1 to n_j , $j = 1,2,\cdots,k$. A finite set of the data is defined as

$$Da := \{d_{i_1,i_2,\cdots,i_k} \big| \ 1 \le i_j \le n_j, j = 1, 2, \cdots, k\} , \tag{2.8}$$

where $k = \dim(Da)$ denotes the dimension of the data set *Da*, or *Da* is a set of *k*-dimensional data. Then, the number of all data in Da will be $N = n_1 n_2 \cdots n_k$.

Definition 2.5 (**Vector expression of multi-dimensional data**) [137]: If the data can be labeled by indices $i_1, i_2, \cdots, i_k$ and arranged in the order of $d_{p_1,\cdots,p_k} < d_{q_1,\cdots,q_k}$, a set of *k*-dimensional data can be arranged as a vector with an ordered *k*-index $Id(i_1, i_2, \cdots, i_k; n_1, n_2, \cdots, n_k)$. Then there exists an integer *r*: $1 \le r \le k$, such that $p_i = q_i$ and $p_r < q_r$, $i < r$.

Consider a set of *N* data with $N = \prod_{i=1}^{k} n_i$. A single index can be used to label the data such as

$$Da = [x_1, x_2, \cdots, x_N] ,$$

then, the data can also be represented using a multi-index $Id(i_1, i_2, \cdots, i_k;\ n_1, n_2, \cdots, n_k)$:

$$Da = [x_{11\cdots1}, x_{11\cdots2}, x_{11\cdots n_k}, \cdots, x_{n_1 n_2 \cdots n_k}] .$$

Example 2.1:

Let $Da = \{x_{ijk} \big| \ 1 \le i \le 2;\ 1 \le j \le 3;\ 1 \le k \le 4\}$. It can be arranged by the multi-indexes $Id(i, j, k; 2, 3, 4)$ such as

$$[x_{111}, x_{112}, x_{113}, x_{114}, x_{121}, x_{122}, x_{123}, x_{124}, x_{131}, x_{132}, x_{133}, x_{134},$$
$$x_{211}, x_{212}, x_{213}, x_{214}, x_{221}, x_{222}, x_{223}, x_{224}, x_{231}, x_{232}, x_{233}, x_{234}].$$

If it is arranged by the multi-index $Id(j, i, k;\ 3, 2, 4)$, then

$$[x_{111}, x_{112}, x_{113}, x_{114}, x_{211}, x_{212}, x_{213}, x_{214}, x_{121}, x_{122}, x_{123}, x_{124},$$
$$x_{221}, x_{222}, x_{223}, x_{224}, x_{131}, x_{132}, x_{133}, x_{134}, x_{231}, x_{232}, x_{233}, x_{234}].$$

If it is arranged by the multi-index $Id(k, j, i;\ 4,3,2)$, then

$$[x_{111}, x_{211}, x_{121}, x_{221}, x_{131}, x_{231}, x_{112}, x_{212}, x_{122}, x_{222}, x_{132}, x_{232},$$
$$x_{113}, x_{213}, x_{123}, x_{223}, x_{133}, x_{233}, x_{114}, x_{214}, x_{124}, x_{224}, x_{134}, x_{234}].$$

Remarks: Generally, the ordered multi-index $Id(i_1, i_2, \cdots, i_k;\ n_1, n_2, \cdots, n_k)$ of one set of data is not unique; it can be expressed by a column vector, or a row vector, or a matrix whose column and row are composed of different ordered indexes.

2.2.2 Arrangement and matrix expression of multi-dimensional data

In this subsection, a set of multi-dimensional data will be arranged into a matrix form. Assume that $Id(i_1, i_2, \cdots, i_k;\ n_1, n_2, \cdots, n_k)$ has multi-index $\{i_{j_1}, i_{j_2}, \cdots, i_{j_p}\} \subset \{i_1, i_2, \cdots, i_k\}$, then, $Id(i_{j_1}, i_{j_2}, \cdots, i_{j_p};\ n_{j_1}, n_{j_2}, \cdots, n_{j_p})$ will be a sub-index of $Id(i_1, i_2, \cdots, i_k;\ n_1, n_2, \cdots, n_k)$.

Definition 2.6 (Matrix expression of multi-dimensional data) [137]: Assume that *Da* is a *k*-dimensional data with $N = i_1 \times i_2 \times \cdots \times i_k = (n_{\beta_1} \cdots n_{\beta_q}) \times (n_{\alpha_1} \cdots n_{\alpha_p})$, and there are two disjoint sub-indices $\{i_{\alpha_1}, i_{\alpha_2}, \cdots, i_{\alpha_p}\} \bigcup \{i_{\beta_1}, i_{\beta_2}, \cdots, i_{\beta_q}\} = \{i_1, i_2, \cdots, i_k\}$ and $\{i_{\alpha_1}, i_{\alpha_2}, \cdots, i_{\alpha_p}\} \bigcap \{i_{\beta_1}, i_{\beta_2}, \cdots, i_{\beta_q}\} = \phi$, which forms a partition of the index of *Da*. Then, *Da* can be expressed as

$$Da = \{\gamma_{11\cdots11},\ \gamma_{11\cdots12}, \cdots, \gamma_{11\cdots1n_{\beta q}},\ \gamma_{11\cdots21},\ \gamma_{11\cdots22}, \cdots,\ \gamma_{11\cdots n_{\beta(q-1)}n_{\beta q}},\ \gamma_{1n_{\alpha2}\cdots n_{\alpha p}n_{\beta1}\cdots n_{\beta q}}, \cdots,\ \gamma_{n_{\alpha1}n_{\alpha2}\cdots n_{\alpha p}n_{\beta1}\cdots n_{\beta q}}\}, \tag{2.9}$$

Da can be arranged into a matrix in the order of

$$Id(i_{\beta_1}, i_{\beta_2}, \cdots, i_{\beta_q};\ n_{\beta_1}, n_{\beta_2}, \cdots, n_{\beta_q}) \times Id(i_{\alpha_1}, i_{\alpha_2}, \cdots, i_{\alpha_p};\ n_{\alpha_1}, n_{\alpha_2}, \cdots, n_{\alpha_p}).$$

For M_{Da} in Eq. (2.10), its rows are vectors labeled by multi-index $Id(i_{\beta_1}, i_{\beta_2}, \cdots, i_{\beta_q};\ n_{\beta_1}, n_{\beta_2}, \cdots, n_{\beta_q})$, and its columns are vectors labeled by multi-index $Id(i_{\alpha_1}, i_{\alpha_2}, \cdots, i_{\alpha_p};\ n_{\alpha_1}, n_{\alpha_2}, \cdots, n_{\alpha_p})$.

$$M_{Da} = \begin{bmatrix} \gamma_{1\ldots11}^{1\cdots11} & \gamma_{1\ldots11}^{1\cdots12} & \cdots & \gamma_{1\ldots11}^{1\cdots1n_{\alpha_p}} & \cdots & \gamma_{1\ldots11}^{n_{\alpha_1}\cdots n_{\alpha_p}} \\ \gamma_{1\ldots12}^{1\cdots11} & \gamma_{1\ldots12}^{1\cdots12} & \cdots & \gamma_{1\ldots12}^{1\cdots1n_{\alpha_p}} & \cdots & \gamma_{1\ldots12}^{n_{\alpha_1}\cdots n_{\alpha_p}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_{n_{\beta_1}\ldots n_{\beta_q}}^{1\cdots11} & \gamma_{n_{\beta_1}\ldots n_{\beta_q}}^{1\cdots12} & \cdots & \gamma_{n_{\beta_1}\ldots n_{\beta_q}}^{1\cdots1n_{\alpha_p}} & \cdots & \gamma_{n_{\beta_1}\ldots n_{\beta_q}}^{n_{\alpha_1}\cdots n_{\alpha_p}} \end{bmatrix} \in \Re^{(n_{\beta_1}\cdots n_{\beta_q})\times(n_{\alpha_1}\cdots n_{\alpha_p})}. \qquad (2.10)$$

With a comparison of the entry in the vector expression in Eq. (2.9), $\gamma_{i_{\beta_1},i_{\beta_2},\cdots,i_{\beta_q}}^{i_{\alpha_1},i_{\alpha_2},\cdots,i_{\alpha_p}} = \gamma_{i_{\alpha_1},i_{\alpha_2},\cdots,i_{\alpha_p},i_{\beta_1},i_{\beta_2},\cdots,i_{\beta_q}}$ is the entry of M_{Da} with $i_{\alpha_1} \in \{1, 2, \cdots, n_{\alpha_1}\}, \cdots$, $i_{\alpha_p} \in \{1, 2, \cdots, n_{\alpha_p}\}$ and $i_{\beta_1} \in \{1, 2, \cdots, n_{\beta_1}\}, \cdots, i_{\beta_q} \in \{1, 2, \cdots, n_{\beta_q}\}$.

Example 2.2:

For the data set $Da = \{x_{ijk} \mid 1 \le i \le 2;\ 1 \le j \le 3;\ 1 \le k \le 4\}$ in Example 2.1, we can re-arrange it into a matrix with multi-index $Id(k;\ 4) \times Id(i,\ j;\ 2, 3)$ such that

$$\begin{bmatrix} x_1^{11} & x_1^{12} & x_1^{13} & x_1^{21} & x_1^{22} & x_1^{23} \\ x_2^{11} & x_2^{12} & x_2^{13} & x_2^{21} & x_2^{22} & x_2^{23} \\ x_3^{11} & x_3^{12} & x_3^{13} & x_3^{21} & x_3^{22} & x_3^{23} \\ x_4^{11} & x_4^{12} & x_4^{13} & x_4^{21} & x_4^{22} & x_4^{23} \end{bmatrix} \in \Re^{4\times6},$$

where $x_k^{ij} = x_{ijk}$, $i = 1, 2;\ j = 1, 2, 3; k = 1, 2, 3, 4$.

Example 2.3:

If *Da* is a 5-dimensional data set with $N = 1\times2\times3\times4\times5 = 120$, and there are two disjoint sub-indices forming a partition of the index set of *Da* as $\{i_1, i_2, i_3\} \cup \{i_4, i_5\} = \{i_1, i_2, \cdots, i_5\}$, then, according to Definition 2.5, *Da* can be expressed as the following vector

$$Da = [x_{11111}, x_{11112}, \cdots, x_{11\cdots5}, \cdots\cdots, x_{12111}, x_{12112}, \cdots, x_{12345}] \in \Re^{1\times120}.$$

At the same time, according to Definition 2.6, *Da* can be arranged into matrices with different orders of 5-dimensional factor $Id\{i_1, i_2, i_3, i_4, i_5\}$, for example,

$$Id(i_4, i_5;\ n_4, n_5) \times Id(i_1, i_2, i_3;\ n_1, n_2, n_3),\ n_1 = 1, n_2 = 2, n_3 = 3, n_4 = 4, n_5 = 5.$$

Then, the matrix M_{Da} can be expressed as

$$M_{Da} = \begin{bmatrix} x_{11}^{111} & x_{11}^{112} & x_{11}^{113} & x_{11}^{121} & x_{11}^{122} & x_{11}^{123} \\ x_{12}^{111} & x_{12}^{112} & x_{12}^{113} & x_{12}^{121} & x_{12}^{122} & x_{12}^{123} \\ & & \cdots & & & \\ x_{15}^{111} & x_{15}^{112} & x_{15}^{113} & x_{15}^{121} & x_{15}^{122} & x_{15}^{123} \\ & & \vdots & & & \\ x_{41}^{111} & x_{41}^{112} & x_{41}^{113} & x_{41}^{121} & x_{41}^{122} & x_{41}^{123} \\ x_{42}^{111} & x_{42}^{112} & x_{42}^{113} & x_{42}^{121} & x_{42}^{122} & x_{42}^{123} \\ & & \cdots & & & \\ x_{45}^{111} & x_{45}^{112} & x_{45}^{113} & x_{45}^{121} & x_{45}^{122} & x_{45}^{123} \end{bmatrix} \in \Re^{(4\times5)\times(1\times2\times3)} = \Re^{20\times6},$$

where the rows of M_{Da} are arranged by multi-index $Id(i_4, i_5;\ n_4, n_5)$, and the columns are arranged by multi-index $Id(i_1, i_2, i_3;\ n_1, n_2, n_3)$.

Comparing the entries in the vector and the matrix of the same matrix *Da* yields

$$x_{i_4, i_5}^{i_1, i_2, i_3} = x_{i_1, i_2, i_3, i_4, i_5},$$

where $i_1 = 1$, $i_2 \in \{1, 2\}$, $i_3 \in \{1, 2, 3\}$, $i_4 \in \{1, 2, 3, 4\}$, $i_5 \in \{1, 2, 3, 4, 5\}$.

2.2.3 Product operators of matrices

A brief review of conventional matrix products will be given first.

(1) "$\otimes$" is the **Kronecker product** of matrices [12], which is also called the tensor product. Specifically, if $A = (a_{ij}) \in \Re^{m\times n}$ and $B = (b_{ij}) \in \Re^{p\times q}$, then the Kronrcker product of *A* and *B* is defined as

$$A \otimes B = \begin{bmatrix} a_{11} \times B & a_{12} \times B & \cdots & a_{1n} \times B \\ a_{21} \times B & a_{22} \times B & \cdots & a_{2n} \times B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} \times B & a_{m2} \times B & \cdots & a_{mn} \times B \end{bmatrix} \in \Re^{mp\times nq}. \tag{2.11}$$

Given a constant $\alpha \in \Re$, the Kronrcker product satisfies

$$\alpha \otimes A = A \otimes \alpha = \alpha A = (\alpha \times a_{ij}) \in \Re^{m\times n}. \tag{2.12}$$

(2) "$\circ$" is **Hadamard product** of matrices [12]. If $A = (a_{ij})$, $B = (b_{ij}) \in \Re^{m\times n}$, then the Hadamard product of *A* and *B* is defined as

$$A \circ B = (a_{ij} b_{ij}) \in \Re^{m \times n}. \quad (2.13)$$

(3) "$*$" is **Khatri-Rao Product** of matrices [12]. Specifically, if $A = (a_{ij}) \in \Re^{m \times r}$, $B = (b_{ij}) \in \Re^{n \times r}$, the Khatri-Rao product of *A* and *B* will be

$$A * B = [Col_1(A) \otimes Col_1(B), Col_2(A) \otimes Col_2(B), \cdots, Col_r(A) \otimes Col_r(B)] \in \Re^{mn \times rr}. \quad (2.14)$$

where $\otimes$ is the Kronecker product, and $Col_i(A)$ and $Col_i(B)$ are the *i*-th columns of *A* and *B*, respectively, $i = 1, 2, \cdots, r$.

2.3 Definitions of the Semi-Tensor Product of Matrices

2.3.1 Left STP of two matrices with multiplier dimension

As a generalization of the conventional matrix products, STP operation of matrices remains the major properties of the conventional matrix product. Moreover, via the swap matrix [137], the STP has some unique pseudo-commutative properties over the conventional matrix product.

Let $A = (a_{ij}) \in \Re^{m \times n}$ and $B = (b_{ij}) \in \Re^{p \times q}$, or $A = (a_{ij}) \in D^{m \times n}$ and $B = (b_{ij}) \in D^{p \times q}$:

(1) If $n = p$, *A* and *B* are said to be of "equal dimension";

(2) If $n = tp$ or $nt = p$, *A* and *B* are said to be of "multiplier dimension", where *t* is a positive integer. In representation, $A \succ_t B$ if $n = tp$, and $A \prec_t B$ if $nt = p$;

(3) Otherwise, *A* and *B* are said to be "arbitrary dimension".

Based on the above conventions, the STP is defined as follows.

Definition 2.7 (Left STP of vectors) [137]:

(1) If $X \in \Re^{1 \times mn}$ is a row and $Y \in \Re^{m \times 1}$ is a column, then, *X* can be split into *m* equal-size blocks as $(X^1 \quad X^2 \quad \cdots \quad X^m)$, such that $X^i \in \Re^{1 \times n}$, $i = 1, 2, \cdots, m$. The left STP of *X* and *Y* can be defined as

$$X \rhd Y := \sum_{i=1}^{m} X^i y_i \in \Re^{1 \times n}. \quad (2.15)$$

(2) If $X \in \Re^{1\times m}$ is a row vector and $Y \in \Re^{mn\times 1}$ is a column vector, then, the left STP of *X* and *Y* can be defined as

$$X \triangleright Y := (Y^T \triangleright X^T)^T \in \Re^{n\times 1}. \tag{2.16}$$

By the STP operation, it can generalize the matrix expression from a bilinear function to a multi-linear function. This thesis mainly focuses on processing the left STP of the multiple-dimension matrices, rather than the STP of arbitrary matrices.

Example 2.4:

(1) If $X = [1 \quad 3 \quad 2 \quad 4]$ and $Y = [2 \quad -1]^T$, the left STP of *X* and *Y* will be

$$X \triangleright Y = [1 \quad 3]\times 2 + [2 \quad 4]\times(-1) = [0 \quad 2].$$

(2) If $X = [1 \quad 2 \quad -1]$ and $Y = [2 \quad 1 \quad -1 \quad 0 \quad -2 \quad 1]^T$, the left STP of *X* and *Y* will be

$$X \triangleright Y = 1\times\begin{bmatrix}2\\1\end{bmatrix} + 2\times\begin{bmatrix}-1\\0\end{bmatrix} + (-1)\times\begin{bmatrix}-2\\1\end{bmatrix} = \begin{bmatrix}2\\0\end{bmatrix}.$$

Definition 2.8 (Left STP of matrices with multiplier dimension) [137]: If $A = (a_{ij}) \in \Re^{m\times n}$, $B = (b_{ij}) \in \Re^{p\times q}$, and $A \succ_t B$, then the left STP of *A* and *B* is defined as

$$A \triangleright B = \begin{bmatrix} Row_1(A) \triangleright Col_1(B) & Row_1(A) \triangleright Col_2(B) & \cdots & Row_1(A) \triangleright Col_q(B) \\ Row_2(A) \triangleright Col_1(B) & Row_2(A) \triangleright Col_2(B) & \cdots & Row_2(A) \triangleright Col_q(B) \\ \vdots & \vdots & \vdots & \vdots \\ Row_m(A) \triangleright Col_1(B) & Row_m(A) \triangleright Col_2(B) & \cdots & Row_m(A) \triangleright Col_q(B) \end{bmatrix} \in \Re^{m\times(tq)}, \tag{2.17}$$

where $t = \frac{n}{p}$, $Row_i(A)$ is the *i*-th row of *A*, $Col_j(B)$ is the *j*-th column of *B*, $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, q$.

Example 2.5:

If $X = \begin{bmatrix} 1 & 2 & -1 & 2 \\ 0 & 1 & 2 & 3 \\ 3 & 3 & 1 & 1 \end{bmatrix}$, $Y = \begin{bmatrix} 1 & 2 \\ -1 & 3 \end{bmatrix}$, then,

$$X \rhd Y = \begin{bmatrix} 1\times(1 \;\; 2) - 1\times(-1 \;\; 2) & 2\times(1 \;\; 2) + 3\times(-1 \;\; 2) \\ 1\times(0 \;\; 1) - 1\times(2 \;\; 3) & 2\times(0 \;\; 1) + 3\times(2 \;\; 3) \\ 1\times(3 \;\; 3) - 1\times(1 \;\; 1) & 2\times(3 \;\; 3) + 3\times(1 \;\; 1) \end{bmatrix} = \begin{bmatrix} 2 & 0 & -1 & 10 \\ -2 & -2 & 6 & 11 \\ 2 & 2 & 9 & 9 \end{bmatrix}.$$

It is obviously seen that if $n = p$ in the above definition, the STP becomes the conventional product of matrices, or

$$A \rhd B = AB. \tag{2.18}$$

Thus the left STP is a generalization of the conventional matrix product. To simplify representation, the symbol "$\rhd$" is omitted in the following chapters unless specified separately.

2.3.2 General STP of matrices with arbitrary dimension

Following the definition strategy of the left STP, the right STP and general STP over arbitrary matrices will be discussed in this subsection.

Recall the Kronecker product of matrices. If $A = (a_{ij}) \in \Re^{m\times n}$ and $B = (b_{ij}) \in \Re^{p\times q}$, then the Kronecker product $A \otimes B = (A \otimes I_p)(I_n \otimes B)$, where I_p and I_n are identity matrices. Hence, the left STP has the following alternative definition

$$A \rhd B = \begin{cases} (A \otimes I_t)B, & if \quad A \prec_t B \\ A(B \otimes I_t), & if \quad A \succ_t B. \end{cases} \tag{2.19}$$

If A and B are of multiplier dimension, the matching right identity matrix can be obtained. Then, the right STP of A and B can be defined as follows.

Definition 2.9 (Right STP of matrices with multiplier dimension) [137]: Given two matrices A and B, and if either $A \prec_t B$ or $A \succ_t B$, then the right STP of A and B, denoted by $A \lhd B$, is defined as

$$A \lhd B = \begin{cases} (I_t \otimes A)B, & if \quad A \prec_t B \\ A(I_t \otimes B), & if \quad A \succ_t B. \end{cases} \tag{2.20}$$

In order to realize the matrix expression for a multi-valued logic, especially for multi-variable fuzzy logic relations, the STP algorithms in Definitions 2.7-2.9 will be extended from the conventional matrix operation to the logic matrix operation, based on the following definitions and algorithms:

Definition 2.10 (Left STP of logic matrices with multiplier dimension): Given two arbitrary logic matrices with multiplier dimension, $A=(a_{ij})\in\Re^{m\times n}$ and $B=(b_{ij})\in\Re^{p\times q}$, the left STP of logic matrices can be defined as

$$A \rhd B=\begin{cases} A\wedge\left(B\otimes I_{t_1}\right)\in\Re^{m\times(t_1 q)}, & \text{if } n/p=t_1 \\ \left(A\otimes I_{t_2}\right)\wedge B\in\Re^{(mt_2)\times q}, & \text{if } p/n=t_2, \end{cases} \tag{2.21}$$

where I_{t_1} and I_{t_2} are identity matrices; $\otimes$ and $\wedge$ are the conventional Kronecker product and *T*-norm operator, respectively.

Definition 2.11 (Right STP of logic matrices with multiplier dimension): Given two arbitrary logic matrices with multiplier dimension $A=(a_{ij})\in\Re^{m\times n}$ and $B=(b_{ij})\in\Re^{p\times q}$, the right STP of logic matrices can be defined as

$$A \lhd B=\begin{cases} \left(I_{t_1}\otimes A\right)\wedge B\in\Re^{(mt_1)\times q}, & \text{if } p/n=t_1 \\ A\wedge\left(I_{t_2}\otimes B\right)\in\Re^{m\times(t_2 q)}, & \text{if } n/p=t_2, \end{cases} \tag{2.22}$$

where I_{t_1} and I_{t_2} are identity matrices; $\otimes$ and $\wedge$ are the conventional Kronecker product and *T*-norm operator, respectively.

Furthermore, if *a* and *b* are two positive integers $a,\ b\in \mathrm{Z}^{+}$, denote the least common multiple of *a* and *b* by $lcm\{a,b\}$.

Definition 2.12 (STPs of two arbitrary matrices) [137]: Given two matrices $A=(a_{ij})\in\Re^{m\times n}$, $B=(b_{ij})\in\Re^{p\times q}$, and $\alpha=lcm\{n,p\}$ where "lcm" denotes the least common multiple operation, then

(1) The general left STP of *A* and *B* is defined as

$$A \rhd B=(A\otimes I_{\alpha/n})\,(B\otimes I_{\alpha/p})\,; \tag{2.23}$$

(2) The general right STP of *A* and *B* is defined as

$$A \lhd B = (I_{\alpha/n} \otimes A)\,(I_{\alpha/p} \otimes B)\,. \tag{2.24}$$

When $n = p$, the left and right STP of matrices become the conventional product of matrices. If $lcm\{n, p\} = n$ or $lcm\{n, p\} = p$, the left STP becomes Definition 2.8, and the right STP is compatible with Definition 2.9. Based on STP algorithms of Eq. (2.23) and Eq. (2.24) in Definition 2.12, the definitions of the STP of two arbitrary logic matrices can be made:

Definition 2.13 (General STP of two arbitrary logic matrices): Given two arbitrary logic matrices $A = (a_{ij}) \in \Re^{m\times n}$ and $B = (b_{ij}) \in \Re^{p\times q}$, the STP of *A* and *B* can be defined as

(1) The general left STP of *A* and *B*:

$$A \rhd B = (A \otimes I_{\alpha/n}) \wedge (B \otimes I_{\alpha/p})\,; \tag{2.25}$$

(2) The general right STP of *A* and *B*:

$$A \lhd B = (I_{\alpha/n} \otimes A) \wedge (I_{\alpha/p} \otimes B)\,, \tag{2.26}$$

where $\alpha = lcm\{n, p\}$; $I_{\alpha/n}$ and $I_{\alpha/p}$ are identity matrices; $\otimes$ and $\wedge$ are the conventional Kronecker product and *T*-norm operator, respectively.

2.3.3 Properties of the STP

The STP of matrices is a generalization of the conventional matrix product. The general STP algorithms have similar properties as those in the conventional matrix product; details can be found in [137].

Consider two general properties of associative law and distributive law as examples for illustration. We will show that the left STP also satisfies these two laws.

Theorem 2.1 [137]: Assume that dimensions of the matrices *A*, *B* and *C* in the following equations meet the dimension requirements (e.g., the STP operator $\rhd$). Then, we have

(1) Distributive Law

$$\begin{cases} A \rhd (aB \pm bC) = aA \rhd B \pm bA \rhd C \\ (aA \pm bB) \rhd C = aA \rhd C \pm bB \rhd C, \end{cases} \quad a, b \in \Re\,. \tag{2.27}$$

(2) Associative Law

$$(A \triangleright B) \triangleright C = A \triangleright (B \triangleright C) . \tag{2.28}$$

The proofs of Eq. (2.27) and Eq. (2.28) can be found in [137].

2.4 Concluding Remarks

In this chapter, some related definitions and preliminaries are introduced first, and then, the measures of multi-dimensional data are defined. The definitions and basic properties of the STP of matrices are given. Finally, the STP algorithms of logic matrices are defined to process conventional logic matrices. Based on the defined mathematical concepts and tools, the matrix expression for specific fuzzy logic and fuzzy reasoning will be discussed in the following chapters. The related contribution in this chapter has been published in:

(1) H. Lyu, W. Wang, X. Liu, "Modeling of multi-variable fuzzy systems by semi-tensor product", *IEEE Transactions on Fuzzy Systems*, Vol. 28, No. 2, pp. 228-235, Feb. 2020.

Chapter 3 Formulation of the Fuzzy Relation Matrices

In the multi-valued logic variables, when $k \to \infty$ for the discrete set D_k, the continuous interval set $D_\infty = \{\alpha \mid 0 \le \alpha \le 1\} = [0,\ 1] \subset \Re$ will be obtained from D_k. When the fuzzy sets are discrete, their ranges still can be described by the multi-valued logic set D_k, but they may also be non-discrete logic sets such as fuzzy sets with different membership functions (MFs) of a fuzzy variable. This chapter will extend the STP concepts of logic matrices into the fuzzy logic inference, multi-variable fuzzy relations, and vector and matrix representations.

In this chapter, firstly, the fundamental representation of the fuzzy logic and fuzzy operations will be reviewed in Section 3.1. The vector expression of fuzzy variables and multi-variable fuzzy logic relations will be developed in Section 3.2. Matrix expressions of multi-variable fuzzy logic relations will be proposed in Section 3.3. Finally, the representation of fuzzy relation matrix (FRM) will be constructed in Section 3.4 for multi-variable fuzzy logic rules.

3.1 Fundamentals of the Related Fuzzy Logic Concepts

This section will summarize the related mathematical preliminaries about fuzzy sets, fuzzy variables, fuzzy relation, and fuzzy reasoning operations using the STP algorithm and the FRM representation.

Definition 3.1 (Fuzzy sets) [142]: A set *X* is a fuzzy set over a universe of discourse *E*, if for a variable $x \in E$, there exits an MF grade $\mu_X(x) \in [0,\ 1]$.

- If $E = \{x_1, \cdots, x_n\}$ is a finite discrete set, then *X* can be expressed as

$$X = \frac{\gamma_1}{x_1} + \frac{\gamma_2}{x_2} + \cdots + \frac{\gamma_n}{x_n}, \tag{3.1}$$

where $\gamma_i = \mu_X(x_i) \in [0,\ 1]$ is the MF grade of x_i on the fuzzy set *X*, $i = 1, \cdots, n$.

- If *E* is a continuous set, then continuous fuzzy set *X* over *E* can be expressed as

$$X = \int_E \frac{\mu_X(x)}{x}. \tag{3.2}$$

Generally, if a fuzzy set *X* is convex and there exists $\mu_X(x) = 1$ for at least one element $x \in E$, then, *X* is normal, and *X* is usually called a fuzzy number [142].

Moreover, if *x* takes a fuzzy number with the corresponding MFs over a universe of discourse, then, *x* is a fuzzy variable.

3.1.1 Review of fuzzy logic operations

According to the discussion in Chapter 2, a *k*-valued logic operation is equivalent to a conventional fuzzy logic operation when it can be processed as a multi-valued logic. Firstly a brief summary is given abut basic fuzzy logic inference operations. If *E* is the universe of discourse for the fuzzy variable *x*, X_1 and X_2 are two fuzzy sets defined over *E*, $\mu(x)$ denotes an MF grade, then

(1) Fuzzy logic complement is the negation operator "NOT" $\neg$, represented by:

$$\neg X: \quad \mu_{\neg X}(x) = 1 - \mu_X(x),\ \forall x \in E. \tag{3.3}$$

(2) Fuzzy logic union is the disjunction "OR" $\bigcup$, which is usually realized as an *S*-norm operator such that:

$$X_1 \bigcup X_2: \quad \mu_{X_1 \cup X_2}(x) = \mu_{X_1}(x) \vee \mu_{X_2}(x),\ \forall x \in E. \tag{3.4}$$

(3) Fuzzy logic intersection is the conjunction "AND" $\bigcap$, which is usually realized as a *T*-norm algorithm:

$$X_1 \bigcap X_2: \quad \mu_{X_1 \cap X_2}(x) = \mu_{X_1}(x) \wedge \mu_{X_2}(x),\ \forall x \in E. \tag{3.5}$$

where "$\vee$" and "$\wedge$" can be chosen as an S-norm operator and a T-norm operator, respectively. As an example, if T-norm is chosen as "min" and S-norm is chosen as "max", then, Eq. (3.4) and (3.5) become

$$X_1 \bigcup X_2: \ \mu_{X_1 \cup X_2}(x) = \max(\mu_{X_1}(x), \mu_{X_2}(x)), \ \forall x \in E, \tag{3.6}$$

$$X_1 \bigcap X_2: \ \mu_{X_1 \cap X_2}(x) = \min(\mu_{X_1}(x), \mu_{X_2}(x)), \ \forall x \in E. \tag{3.7}$$

3.1.2 Review of fuzzy relations

Definition 3.2 [143]: Assume E_i is a universe of discourse for a fuzzy variable x_i, $i = 1, 2, \cdots, k$. A fuzzy relation R for all x_i over E_i, $i = 1, 2, \cdots, k$, is a fuzzy set on the Cartesian product space $\prod_{i=1}^{k} E_i = E_1 \times \cdots \times E_k$. Specifically, for each point $(x_1, \cdots, x_k)$ $\in \prod_{i=1}^{k} E_i$, there is an MF grade

$$\mu_R(x_1, \cdots, x_k) = \wedge \{\mu_{E_1}(x_1), \cdots, \mu_{E_k}(x_k)\}, \tag{3.8}$$

where $x_i \in E_i$ and $\mu_{E_i}(x_i)$ is its MF grade, $i = 1, 2, \cdots, k$.

In particular, consider two universes of discourse E_1 and E_2. A fuzzy relation R: $E_1 \times E_2 \to [0,\ 1]$ is defined on the two-dimensional Cartesian product space $E_1 \times E_2$ with "$\wedge = \min$" by

$$\mu_R(x_1, x_2) = \min\{\mu_{E_1}(x_1), \mu_{E_2}(x_2)\}, \ \forall \ \ x_1 \in E_1, \ \ x_2 \in E_2. \tag{3.9}$$

In conventional fuzzy theory, the fuzzy relation between two fuzzy variables can be described by a matrix if both variables are defined on the finite discrete universes of discourse. For example, given $E_1 = \{x_{11}, \cdots, x_{1n}\}$ and $E_2 = \{x_{21}, \cdots, x_{2m}\}$, the fuzzy relation R over $E_1 \times E_2$ can be expressed by

$$\mu_R(x_{1i}, x_{2j}) = \min\{\mu_{E_1}(x_{1i}), \mu_{E_2}(x_{2j})\}, \ i = 1, \ldots, n; j = 1, \ldots, m,$$

which has the following relation matrix between x_1 and x_2:

$$M_R = \begin{bmatrix} \mu_R(x_{11}, x_{21}) & \mu_R(x_{11}, x_{22}) & \cdots & \mu_R(x_{11}, x_{2m}) \\ \mu_R(x_{12}, x_{21}) & \mu_R(x_{12}, x_{22}) & \cdots & \mu_R(x_{12}, x_{2m}) \\ & \vdots & & \\ \mu_R(x_{1n}, x_{21}) & \mu_R(x_{1n}, x_{22}) & \cdots & \mu_R(x_{1n}, x_{2m}) \end{bmatrix}.$$

The fuzzy composition operation between two fuzzy relations will be introduced in the next subsection.

3.1.3 Conventional fuzzy composition operations

If E_1, E_2, E_3 are three real sets, assume that R_1 and R_2 are fuzzy relations defined on Cartesian product spaces $E_1 \times E_2$ and $E_2 \times E_3$, respectively. Then, $R_3 = R_1 \circ R_2$ is a fuzzy composition of the relations R_1 and R_2 on the Cartesian product space $E_1 \times E_3$ [144], whose MF grade can be determined by

$$R_3 = R_1 \circ R_2 :\ \mu_{R_3}(x_1, x_3) = \underset{x_2 \in E_2}{\vee} \{\mu_{R_1}(x_1, x_2) \wedge \mu_{R_2}(x_2, x_3)\},\ x_1 \in E_1,\ x_3 \in E_3, \quad (3.10)$$

where "$\vee$" and "$\wedge$" can be chosen as an S-norm operator and a T-norm operator, respectively. For example, if $\vee$ is chosen as "max" and $\wedge$ is chosen as "min", then

$$\mu_{R_3}(x_1, x_3) = \max_{x_2 \in E_2} \{\min(\mu_{R_1}(x_1, x_2), \mu_{R_2}(x_2, x_3))\},\ x_1 \in E_1,\ x_3 \in E_3. \quad (3.11)$$

Specifically, if the universes of discourse for all variables are finite discrete, such as

$$E_1 = \{x_{11}, \cdots, x_{1n}\},\quad E_2 = \{x_{21}, \cdots, x_{2m}\},\quad E_3 = \{x_{31}, \cdots, x_{3k}\}, \quad (3.12)$$

then, the corresponding MF grade of the composition relation can be described by:

$$\mu_{R_3}(x_{1i}, x_{3t}) = \overset{m}{\underset{j=1}{\vee}} \{\mu_{R_1}(x_{1i}, x_{2j}) \wedge \mu_{R_2}(x_{2j}, x_{3t})\},\ i = 1, 2, \cdots, n,\ t = 1, 2, \cdots, k. \quad (3.13)$$

Assume that the two-variable relation matrices of R_1, R_2, and R_3 are M_{R_1}, M_{R_2}, and M_{R_3}, respectively. Then,

$$M_{R_3} = M_{R_1} M_{R_2}, \quad (3.14)$$

where the traditional product rule is replaced by a T-norm operation and the traditional addition rule is replaced by an S-norm operation.

For example, in a general family, the genetic relations of genes between two adjacent generations would be similar. For instance, the relation R_1 between "grandpa and grandma" and "father and mother" is similar to the relation R_2 between "father and mother" and "son and daughter". The corresponding two-variable relation matrices will be:

$$M_{R_1} = \begin{array}{c} \\ grandpa \\ grandma \end{array} \begin{array}{c} \begin{array}{cc} father & mother \end{array} \\ \begin{bmatrix} 0.8 & 0.9 \\ 0.7 & 0.8 \end{bmatrix} \end{array}, \text{ and}$$

$$M_{R_2} = \begin{array}{c} \\ father \\ mother \end{array} \begin{array}{c} \begin{array}{cc} son & daughter \end{array} \\ \begin{bmatrix} 0.8 & 0.9 \\ 0.7 & 0.8 \end{bmatrix} \end{array}.$$

If $\vee$ and $\wedge$ are chosen as "max" and "min", respectively, the two-variable relation matrix for the genetic relation R_3 between "grandpa and grandma" and "son and daughter" can be determined by

$$\begin{aligned} M_{R_3} &= M_{R_1} M_{R_2} \\ &= \begin{bmatrix} 0.8 & 0.9 \\ 0.7 & 0.8 \end{bmatrix} \begin{bmatrix} 0.8 & 0.9 \\ 0.7 & 0.8 \end{bmatrix} \\ &= \begin{bmatrix} (0.8 \wedge 0.8) \vee (0.9 \wedge 0.7) & (0.8 \wedge 0.9) \vee (0.9 \wedge 0.8) \\ (0.7 \wedge 0.8) \vee (0.8 \wedge 0.7) & (0.7 \wedge 0.9) \vee (0.8 \wedge 0.8) \end{bmatrix} \\ &= \begin{bmatrix} 0.8 & 0.8 \\ 0.7 & 0.8 \end{bmatrix}. \end{aligned}$$

3.2 Vector Expression of Fuzzy Variables and Fuzzy Relations

A brief description of vector expressions of fuzzy sets, fuzzy variables and fuzzy relations will be provided in this section.

Definition 3.3 (Vector expression of fuzzy sets): Consider a universe of discourse *E*. By Definition 3.1, *X* is a fuzzy set over *E* with MF grade $\mu_X(x) = \gamma_x \in [0,\ 1]$, *X* can be expressed by the following vector:

- If *E* is a finite discrete set with $E = \{x_1, \cdots, x_n\}$, then *X* can be expressed as

$$V_E(X) := (\gamma_1 \ \ \cdots \ \ \gamma_n)^T \in \Re^{n\times 1}, \tag{3.15}$$

where $\gamma_i = \mu_X(x_i) \in [0,\ 1]$ is the MF grade of x_i on the fuzzy set *X*, $x_i \in E$, $i = 1, \cdots, n$.

- If *E* is a continuous set, then, the continuous fuzzy set *X* over *E* cannot be expressed by a finite vector.

Definition 3.4 (Vector expression of fuzzy variables): Consider a fuzzy variable *x* over a universe of discourse *E* and a group of fuzzy sets $X_1, \cdots, X_m$ over *E* with its MF grade $\mu_{X_j}(x) \in [0,\ 1]$, $x \in E$, $j = 1, \cdots, m$. A fuzzy variable $x \in E$ can be defined by the following vector:

$$V_E(x) := (\gamma_1 \ \ \cdots \ \ \gamma_m)^T \in \Re^{m\times 1}, \tag{3.16}$$

where $x \in E$, $\gamma_j = \mu_{X_j}(x)$ is the MF grade of *x* on the fuzzy set X_j, $j = 1, \cdots, m$. The universe of discourse *E* can be either a discrete or continuous set.

Definition 3.5 (Vector expression of fuzzy relations): Assume that $X_i^1, \cdots, X_i^{N_i}$ are N_i fuzzy sets defined over a universe of discourse E_i, $i = 1, \cdots, n$. If $V_{E_i}(x_i) = (\mu_{X_i^1}(x_i), \cdots, \mu_{X_i^{N_i}}(x_i))^T$ is a vector expression of a fuzzy variable $x_i \in E_i$, $i = 1, \cdots, n$, then the vector expression of a fuzzy relation $R:\ E_1 \times \cdots \times E_n \to [0,\ 1]$ in Definition 3.2 (Eq. (3.8)) can be determined by

$$V_R(x_1, \cdots, x_n) = V_{E_1}(x_1) \triangleright V_{E_2}(x_2) \triangleright \cdots \triangleright V_{E_n}(x_n) \in \Re^{(N_1 N_2 \cdots N_n)\times 1}. \tag{3.17}$$

If the MF grade is $\gamma_i^j := \mu_{X_i^j}(x_i)$, $j = 1, 2, \cdots, N_i$, then,

$$V_{E_i}(x_i) = (\gamma_i^1, \cdots, \gamma_i^{N_i})^T \in \Re^{N_i \times 1},\ i = 1, 2, \cdots, n.$$

Consider an example of $n = 2$,

$$V_R(x_1, x_2) = V_{E_1}(x_1) \triangleright V_{E_2}(x_2) = (\gamma_1^1, \cdots, \gamma_1^{N_1})^T \triangleright (\gamma_2^1, \cdots, \gamma_2^{N_2})^T$$

$$= (\gamma_1^1 \wedge \gamma_2^1,\ \gamma_1^1 \wedge \gamma_2^2, \cdots, \gamma_1^1 \wedge \gamma_2^{N_2},\ \cdots\ , \gamma_1^{N_1} \wedge \gamma_2^1,\ \gamma_1^{N_1} \wedge \gamma_2^2, \cdots, \gamma_1^{N_1} \wedge \gamma_2^{N_2})^T . \quad (3.18)$$

3.3 Matrix Expression of Multi-Variable Fuzzy Relations

In the traditional fuzzy theory, a matrix expression can only be defined between two discrete fuzzy variables with finite elements over the Cartesian product space. The matrix expression of fuzzy relations cannot be used for either multi-dimensional fuzzy variables or continuous universes of discourse. However, based on the concepts of the multi-dimensional data and STP algorithms as discussed in Chapter 2, the relation matrix for multiple fuzzy variables over continuous universes of discourse will be defined as follows.

Definition 3.6: Assume that $X_i^1, \cdots, X_i^{N_i}$ are fuzzy sets defined over a universe of discourse E_i for a fuzzy variable x_i , $i = 1, 2, \cdots, n$. Separate $\{1, 2, \cdots, n\}$ into two disjoint partitions $\{1, 2, \cdots, r\}$ and $\{r+1, r+2, \cdots, n\}$, $r \in Z^+$, $1 \le r < n$. A fuzzy relation R: $E_1 \times \cdots \times E_n \to [0,\ 1]$ for $(x_1, x_2, \cdots, x_n)$ can be treated as a fuzzy relation R: $(E_1 \times \cdots \times E_r) \times (E_{r+1} \times \cdots \times E_n) \to [0,\ 1]$ for $(x_1, x_2, \cdots, x_r)$ and $(x_{r+1}, x_{r+2}, \cdots, x_n)$, whose multi-variable fuzzy relation matrix (FRM) can be determined by

$$M_R(x_1, \cdots, x_n) = V_{R_2}(x_{r+1}, x_{r+2}, \cdots, x_n) \triangleright V_{R_1}^T(x_1, x_2, \cdots, x_r) , \quad (3.19)$$

where $V_{R_1}(x_1, x_2, \cdots, x_r) = V_{E_1}(x_1) \triangleright V_{E_2}(x_2) \triangleright \cdots \triangleright V_{E_r}(x_r)$ is the vector expression of a fuzzy relation R_1 : $E_1 \times \cdots \times E_r \to [0,\ 1]$, and

$V_{R_2}(x_{r+1}, x_{r+2}, \cdots, x_n) = V_{E_{r+1}}(x_{r+1}) \triangleright V_{E_{r+2}}(x_{r+2}) \triangleright \cdots \triangleright V_{E_n}(x_n)$ is the vector expression of a fuzzy relation R_2 : $E_{r+1} \times \cdots \times E_n \to [0,\ 1]$.

If the MF grade $\gamma_i^j := \mu_{X_i^j}(x_i)$, $j = 1, 2, \cdots, N_i$, then, $V_{E_i}(x_i) = (\gamma_i^1, \cdots, \gamma_i^{N_i}))^T$, $i = 1, 2, \cdots, n$. In particular, for a case with $n = 5$, $r = 2$, $n - r = 3$, $V_{R_1}(x_1, x_2)$ and $V_{R_2}(x_3, x_4, x_5)$ can be determined by

$$V_{R_1}(x_1, x_2) = V_{E_1}(x_1) \triangleright V_{E_2}(x_2)$$

$$= (\gamma_1^1 \wedge \gamma_2^1,\ \gamma_1^1 \wedge \gamma_2^2, \cdots, \gamma_1^1 \wedge \gamma_2^{N_2},\ \cdots\ , \gamma_1^{N_1} \wedge \gamma_2^1,\ \gamma_1^{N_1} \wedge \gamma_2^2, \cdots, \gamma_1^{N_1} \wedge \gamma_2^{N_2})^T$$

$$:= (\alpha_1, \alpha_2, \cdots, \alpha_{M_1})^T,\ M_1 = N_1 N_2, \tag{3.20}$$

and

$$V_{R_2}(x_3, x_4, x_5) = V_{E_3}(x_3) \triangleright V_{E_4}(x_4) \triangleright V_{E_5}(x_5)$$

$$= (\gamma_3^1 \wedge \gamma_4^1 \wedge \gamma_5^1, \gamma_3^1 \wedge \gamma_4^1 \wedge \gamma_5^2, \cdots, \gamma_3^1 \wedge \gamma_4^1 \wedge \gamma_5^{N_5},\ \cdots\ , \gamma_3^{N_3} \wedge \gamma_4^{N_4} \wedge \gamma_5^1, \gamma_3^{N_3} \wedge \gamma_4^{N_4} \wedge \gamma_5^2, \cdots, \gamma_3^{N_3} \wedge \gamma_4^{N_4} \wedge \gamma_5^{N_5})^T$$

$$:= (\beta_1, \beta_2, \cdots, \beta_{M_2})^T,\ M_2 = N_3 N_4 N_5. \tag{3.21}$$

Thus, the FRM of the fuzzy relation $R:\ E_1 \times \cdots \times E_5 \to [0,\ 1]$ for $(x_1, x_2, \cdots, x_5)$ will be

$$V_R(x_1, x_2, x_3, x_4, x_5) = V_{R_2}(x_3, x_4, x_5) \triangleright V_{R_1}^T(x_1, x_2)$$

$$= (\beta_1, \beta_2, \cdots, \beta_{M_2})^T \triangleright (\alpha_1, \alpha_2, \cdots, \alpha_{M_1})$$

$$= \begin{pmatrix} \beta_1 \wedge \alpha_1 & \beta_1 \wedge \alpha_2 & \cdots & \beta_1 \wedge \alpha_{M_1} \\ \beta_2 \wedge \alpha_1 & \beta_2 \wedge \alpha_2 & \cdots & \beta_2 \wedge \alpha_{M_1} \\ & \vdots & & \\ \beta_{M_2} \wedge \alpha_1 & \beta_{M_2} \wedge \alpha_2 & \cdots & \beta_{M_2} \wedge \alpha_{M_1} \end{pmatrix} \in \Re^{(N_3 N_4 N_5) \times (N_1 N_2)}. \tag{3.22}$$

3.4 Matrix Expression of Fuzzy Rules and Fuzzy Reasoning

Generally, in a conventional SISO fuzzy system, the fuzzy rule base can be represented by a two-variable FRM. For fuzzy systems with more than one input variable or more than one output variable, however, it is difficult to express the fuzzy rule base in a matrix form according to the traditional fuzzy logic theory. Using the STP of logic matrices and the Definition 3.6 proposed in Section 3.3, it is possible to represent a MIMO fuzzy logic system by an FRM through the following procedures.

Considering an MIMO fuzzy logic model with the rules:

$$R^l:\ IF\ \ (x_1\ \ is\ \ X_1^{p_1})\ \ AND\cdots AND\ \ (x_n\ \ is\ \ X_n^{p_n}),$$

$$THEN\ \ (y_1\ \ is\ \ Y_1^l)\ \ AND\cdots AND\ \ (y_m\ \ is\ \ Y_m^l),\ l=1,2,\cdots,M\ , \quad (3.23)$$

where x_i is the i-th input variable on the universe of discourse E_i; $X_i^1,\cdots,X_i^{N_i}$ are fuzzy sets defined on E_i; N_i is the number of fuzzy sets, $i=1,2,\cdots,n$, $p_i\in\{1,2,\cdots,N_i\}$, $l=p_1+\sum_{i=2}^{n}[(p_i-1)\prod_{j=1}^{i-1}N_j]$; $M=N_1N_2\cdots N_n$ is the number of fuzzy rules; y_j is the j-th output variable over the universe of discourse F_j; $Y_j^1,\cdots,Y_j^M$ are fuzzy sets defined on F_j, $j=1,2,\cdots,m$.

In general, a fuzzy model with fuzzy rules in Eq. (3.23) mathematically represents a fuzzy relation between input and output variables. The construction of matrix expression of the fuzzy model in Eq. (3.23) is to formulate an FRM between multi-dimensional fuzzy variables $\{x_1,x_2,\cdots,x_n\}$ and $\{y_1,y_2,\cdots,y_m\}$ on the space $\prod_{i=1}^{n}E_i\times\prod_{j=1}^{m}F_j=$ $(E_1\times\cdots\times E_n)\times\ (F_1\times\cdots\times F_m)$.

Definition 3.7: Consider a fuzzy system with rules in Eq. (3.23). A MIMO fuzzy relation $R:\ (E_1\times\cdots\times E_n)\times(F_1\times\cdots\times F_m)\rightarrow[0,\ 1]$ can be determined by

$$\mu_R(x_1,\cdots,x_n;y_j,\cdots,y_m)=\wedge\{\wedge(\mu_{X_1}(x_1),\cdots,\mu_{X_n}(x_n)),\ \ \wedge(\mu_{Y_1}(y_1),\cdots,\mu_{Y_m}(y_m))\}, \quad (3.24)$$

where E_i and F_j are the universes of discourse for x_i and y_j, respectively, $i=1,2,\cdots,n$; $j=1,2,\cdots,m$.

Over the universe of discourse for each input variable x_i, there are N_i fuzzy sets, $X_i^1,\cdots,X_i^{N_i}$, $i=1,2,\cdots,n$. Then, based on Definitions 3.4-3.6, the vector expression of each input variable x_i over E_i can be represented as $V_{E_i}(x_i)=(\mu_{X_i^1}(x_i),\cdots,\mu_{X_i^{N_i}}(x_i))^T$,

$i=1,2,\cdots,n$. The input FRM (or vector expression) for input variables $(x_1,\cdots,x_n)$ can be formulated by

$$V_E(x_1,\cdots,x_n)=V_{E_1}(x_1)\triangleright V_{E_2}(x_2)\triangleright\cdots\triangleright V_{E_n}(x_n). \tag{3.25}$$

Similarly, over the universe of discourse for the output variable y_j, there are M fuzzy sets, $Y_j^1,\cdots,Y_j^M$, $j=1,2,\cdots,m$. The vector expression of each output variable y_j on F_j can be described as $V_{F_j}(y_j)=(\mu_{Y_j^1}(y_j),\cdots,\mu_{Y_j^M}(y_j))^T$, $j=1,2,\cdots,m$. Then, the output FRM (or vector expression) for output variables $(y_1,\cdots,y_m)$ can be formulated as

$$V_F(y_1,\cdots,y_m)=V_{F_1}(y_1)\triangleright V_{F_2}(y_2)\triangleright\cdots\triangleright V_{F_m}(y_m). \tag{3.26}$$

Hence, for an MIMO fuzzy system with fuzzy rules in Eq. (3.23), the FRM for the multi-variable fuzzy relations R on $\prod_{i=1}^{n}E_i\times F_j$ can be formulated by:

$$M_R(x_1,\cdots,x_n;y_1,\cdots,y_m):=V_F(y_1,\cdots,y_m)\triangleright V_E^T(x_1,\cdots,x_n)\in\Re^{M^m\times(N_1\times\cdots\times N_n)}. \tag{3.27}$$

Specifically, when $m=1$, Eq. (3.23) becomes a multi-input and single-output (MISO) fuzzy system with one output variable y_j:

$$R_j^l:\ IF\ (x_1\ is\ X_1^{p_1})\ AND\cdots AND\ (x_n\ is\ X_n^{p_n}),\ THEN\ (y_j\ is\ Y_j^l),$$

$$j\in\{1,2,\cdots,m\},\ l=1,2,\cdots,M. \tag{3.28}$$

Consider a fuzzy system in Eq. (3.28). Similar to Eq. (3.24), an MISO fuzzy relation $R_j:\ E_1\times\cdots\times E_n\times F_j\to[0,\ 1]$ can be determined by

$$\mu_{R_j}(x_1,\cdots,x_n;y_j)=\wedge\{\wedge(\mu_{X_1}(x_1),\cdots,\mu_{X_n}(x_n)),\mu_{Y_j}(y_j)\}, \tag{3.29}$$

where E_i and F_j are the universes of discourse for x_i and y_j, respectively, $i=1,2,\cdots,n$; $j=1,2,\cdots,m$.

The FRM for M MISO fuzzy relations R_j^l on $\prod_{i=1}^{n}E_i\times F_j$ with $l=1,2,\cdots,M$ can be formulated by:

$$M_{R_j}\left(x_1,\cdots,x_n;y_j\right):=V_{F_j}(y_j)\triangleright V_E^T(x_1,\cdots,x_n)\in\Re^{M\times(N_1\times\cdots\times N_n)}. \tag{3.30}$$

Eq. (3.27) and (3.30) can be used to generate the relation matrix for IF-THEN rules in Eq. (3.23) and (3.28) without using conventional fuzzy reasoning operations. Some detailed manipulation will be discussed in Sections 4.4 in Chapter 4.

Given the inputs at one sampling instant $x(t)=(x_1(t),x_2(t),\cdots,x_n(t))$, the vector expression for outputs $y(t)=(y_1(t),y_2(t),\cdots,y_m(t))$ can be calculated from the FRM and $x(t)$, i.e.

$$V_F(y_1(t),y_2(t),\cdots,y_m(t))=M_R\triangleright V_E(x_1(t),x_2(t),\cdots,x_n(t)), \tag{3.31}$$

where $V_E(x_1(t),\cdots,x_n(t))=V_{E_1}(x_1(t))\triangleright V_{E_2}(x_2(t))\triangleright\cdots\triangleright V_{E_n}(x_n(t))$ can be computed from $V_{E_i}(x_i(t))=(\mu_{X_i^1}(x_i(t)),\cdots,\mu_{X_i^{N_i}}(x_i(t)))^T$, $i=1,2,\cdots,n$.

$V_{F_j}(y_j(t))=(\mu_{Y_j^1}(y_j(t)),\cdots,\mu_{Y_j^M}(y_j(t)))^T$, $j=1,2,\cdots,m$, can be obtained from $V_F(y_1(t),\cdots,y_m(t))$ through the inverse operation of Eq. (3.26). Specifically, when $m=1$, $V_F(y_1,\cdots,y_m)=V_{F_1}(y_1)$.

Each output variable $y_j(t)$ in Eq. (3.28) will be computed by defuzzifying $V_{F_j}(y_j(t))$, $j=1,2,\cdots,m$, as illustrated in [113]. By defuzzification, the vector expression of the output in Eq. (3.31) can be converted to real values.

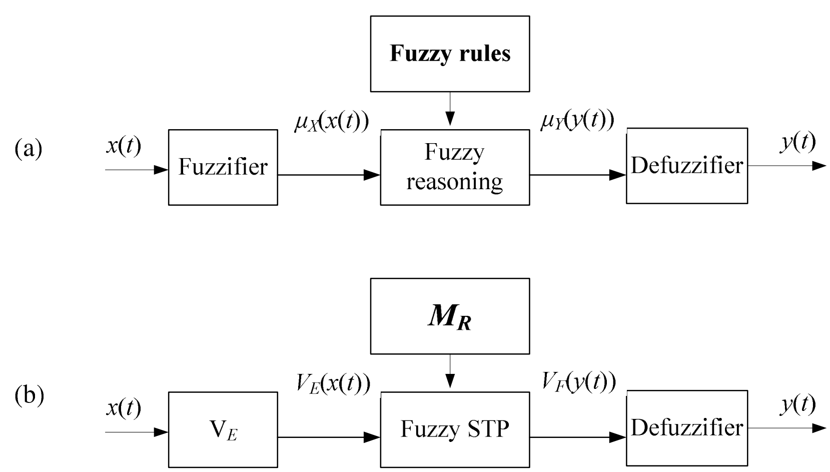


Fig. 3.1. Illustration of two fuzzy systems: (a) the traditional fuzzy reasoning system (b) the proposed FRM reasoning system (to be discussed in Chapter 4).

Based on Definition 3.7, FRM models will be proposed to identify fuzzy systems in Chapter 4. Fig. 3.1(a) illustrates the structure of the traditional fuzzy system. As illustrated in Fig. 3.1(b), the FRMs and fuzzy STP system can replace the fuzzy rules and fuzzy reasoning so that the algebra calculation can be implemented for fuzzy systems.

Moreover, the FRM reasoning can be more suitable for digital implementation based on precise databases than the classical fuzzy linguistic rules and fuzzy logic inference.

3.5 Concluding Remarks

In this chapter, some concepts and definitions in FRM models are introduced, including the fundamental representation of the fuzzy sets, fuzzy relations, and fuzzy reasoning operations, followed by the vector expression and matrix expression of fuzzy variables and fuzzy relations. Then, the construction of FRM models is discussed for MIMO rules-based fuzzy systems, which will be used in the following chapters. The related contribution in this chapter has been published in:

(1) H. Lyu, W. Wang, X. Liu, "Modeling of multi-variable fuzzy systems by semi-tensor product", *IEEE Transactions on Fuzzy Systems*, Vol. 28, No. 2, pp. 228-235, Feb. 2020.

Chapter 4 Construction of Fuzzy Relation Matrix Models

Based on the vector and matrix representation of fuzzy logic operation in Chapter 3, a fuzzy logic relation with multiple fuzzy variables can be expressed by a fuzzy relation matrix (FRM). Then, with the help of semi-tensor product (STP) operation, matrices in fuzzy logic reasoning can be formulated such that fuzzy logic can be expressed as an algebraic equation. In general, the FRM and STP theory assumes that the input-output FRM is known [113]. However, the relation matrices have to be constructed before FRMs are applied. To tackle this problem, the objective of this chapter is to propose a new FRM technique for multi-variable fuzzy system modeling. It is new in the following aspects: 1) A novel system identification technique is proposed to construct an FRM model based on the STP algorithms of logic matrices, by using a direct modeling method. 2) An indirect identification method is proposed to identify FRM models for fuzzy systems.

The rest of this chapter is organized as follows. After a brief overview of the FRM modeling in Section 4.1, Section 4.2 discusses the direct modeling method, and Section 4.3 presents the indirect identification of an FRM model. The effectiveness of the proposed FRM modeling techniques will be tested using some simulation tests in Section 4.4.

4.1 Overview of FRM Modeling

To design a matrix-based fuzzy system, the key step is to construct an FRM model from the available input and output training data.

Given the following input and output training data pairs

$$(x_{1p}^{*},\ x_{2p}^{*},\cdots,x_{np}^{*};\ y_{jp}^{*}),\ p=1,\ 2,\cdots,\ P,\ j=1,\ 2,\cdots,\ m, \tag{4.1}$$

which represents the relation between the input and the output, and P is the number of data pairs. Such a relation can be modeled by several mathematical methods, such as linear regression, neural network, fuzzy logic, and least square estimator (LSE) [145]. In this chapter, the fuzzy logic STP algorithm will be used to identify a model for the input-output relation of the training data. Although the traditional fuzzy logic reasoning can be used to identify the relation, the proposed direct and indirect modeling strategies will identify the input-output fuzzy relation by using the FRM and STP.

The FRM model in Eq. (3.27) can be constructed using the training data in Eq. (4.1), but the accuracy of FRM model will depend on accuracy of the training dataset. Hence, the data in Eq. (4.1) should be selected to cover all possible operating regions in practical system applications. On the other hand, there exist modeling errors for the FRMs inevitably, because the output value of a fuzzy system based on FRMs is an approximation of the actual output variable. Correspondingly, a direct FRM modeling method will be proposed firstly in the next section for system identification.

In the construction of an FRM model, there are many factors that influence modeling accuracy of the identified fuzzy system [146]. Each parameter in a fuzzy inference model will affect its properties and model outputs. Generally, the identification of an FRM model is a process in defining the numbers of fuzzy sets and their membership functions (MFs) and selecting a fuzzy rule base [147, 148]. The FRM modeling accuracy depends on the MFs and the universe of discourse for each fuzzy variable, fuzzy reasoning operations, etc. The RFM parameters can be adjusted manually or automatically through training. If the centers of MFs for output variables are unknown, FRM model can be identified manually, by using a direct modeling method as discussed in the next section.

4.2 Direct FRM Modeling

The direct modeling method will be proposed to construct an FRM model based on the input-output data in Eq. (4.1), which can replace a traditional fuzzy rule base so as to simplify the process of fuzzy reasoning.

The direct FRM model can be formulated based on the following procedures:

1) Specify the universes of discourse for each input variable x_i and each output variable y_j; define N_i fuzzy sets for x_i and M fuzzy sets for y_j, which should cover the whole operating ranges for both x_i and y_j. Specify the fuzzy sets for x_i and y_j as

$$E_{x_i} = \{X_i^1, \cdots, X_i^{N_i}\},\ i = 1, 2, \cdots, n, \tag{4.2}$$

$$E_{y_j} = \{Y_j^1, \cdots, Y_j^M\},\ M = N_1 N_2 \cdots N_n. \tag{4.3}$$

2) Define a complete set of fuzzy rules such as:

$$R_j^l : IF\ (x_1\ is\ X_1^{p_1})\ AND \cdots AND\ (x_n\ is\ X_n^{p_n}),\ THEN\ (y_j\ is\ Y_j^l),\ l = 1, 2, \cdots, M\ , \tag{4.4}$$

where M is the number of fuzzy rule; $p_i \in \{1, 2, \cdots, N_i\}$, $l = p_1 + \sum_{i=2}^{n}[(p_i - 1)\prod_{j=1}^{i-1} N_j]$, $i = 1, 2, \cdots, n$, $j \in \{1, 2, \cdots, m\}$.

3) Specify fuzzy inference operations to implement the STP of logic matrices in the modeling of FRMs. "AND" and "IMPLICATION" can be chosen as a T-norm operator (e.g., product); "OR" can be chosen as an S-norm operator (e.g., max).

4) Determine vector expressions V_{E_i} and V_{F_j}:

$$V_{E_i}(x_{ip}^*) = (\mu_{X_i^1}(x_{ip}^*), \cdots, \mu_{X_i^{N_i}}(x_{ip}^*))^T,\ i = 1, 2, \cdots, n, \tag{4.5}$$

$$V_{F_j}(y_{jp}^*) = (\mu_{Y_j^1}(y_{jp}^*), \cdots, \mu_{Y_j^M}(y_{jp}^*))^T. \tag{4.6}$$

5) The direct input FRM for the p-th training data is formulated as

$$V_E(x_{1p}^*, \cdots, x_{np}^*) = V_{E_1}(x_{1p}^*) \triangleright V_{E_2}(x_{2p}^*) \triangleright \cdots \triangleright V_{E_n}(x_{np}^*). \tag{4.7}$$

As y_j is a single output variable in Eq. (4.4), the output FRM for the p-th data can be determined by

$$V_F(y_{jp}^*) = V_{F_j}(y_{jp}^*) = (\mu_{Y_j^1}(y_{jp}^*), \cdots, \mu_{Y_j^M}(y_{jp}^*))^T. \tag{4.8}$$

6) Construct the p-th FRM model between the p-th input and the p-th output variables:

$$M_{R_j p}\left(x_{1p}^*, x_{2p}^*, \cdots, x_{np}^*; y_{jp}^*\right) = V_{F_j}(y_{jp}^*) \triangleright V_E^T(x_{1p}^*, x_{2p}^*, \cdots, x_{np}^*),$$

$$M_{R_{jp}} \in \Re^{M\times(N_1\times\cdots\times N_n)},\ p=1,2,\cdots,P. \tag{4.9}$$

7) Determine the final FRM model such as:

$$M_{R_j}\left(x_1, x_2, \cdots, x_n; y_j\right) := \bigcup_{p=1}^{P} M_{R_{jp}}\left(x_{1p}^*, x_{2p}^*, \cdots, x_{np}^*; y_{jp}^*\right). \tag{4.10}$$

Eq. (4.10) can be denoted as R_j or M_{R_j} for simplicity; $\bigcup$ can be chosen as any *S*-norm operator, but the "max" operator is used in this work as an example for illustration.

Given an FRM model M_{R_j} and its inputs at one sampling instant $x_i(t)$, $i=1,2,\cdots,n$, the output $y_j(t)$ from the fuzzy system can be determined by the FRM inference operation in Eq. (3.31) and a defuzzification method. Given a pair of training data in Eq. (4.1), $(x_{1p}^*, x_{2p}^*, \cdots, x_{np}^*;\ y_{1p}^*, y_{2p}^*, \cdots, y_{mp}^*)$, $p=1,2,\cdots,P$, a MIMO FRM model can be constructed based on the direct modeling process. The main difference between a multi-input and single-output (MISO) FRM model and an MIMO FRM model is that the output FRM for the *p*-th training data is

$$V_F(y_{1p}^*, \cdots, y_{mp}^*) = V_{F_1}(y_{1p}^*) \triangleright V_{F_2}(y_{2p}^*) \triangleright \cdots \triangleright V_{F_n}(y_{mp}^*),\ p=1,2,\cdots,P. \tag{4.11}$$

4.3 Indirect FRM Modeling

In the direct modeling of an FRM in Section 4.2, the parameters are assumed to be tuned manually. However, if the centers in output fuzzy sets are unknown or uncertain, they also can be adjusted automatically by the LSE algorithm, because they appear as linear parameters in FRM models. Then, an indirect FRM modeling method is proposed for identification of an FRM in this section.

4.3.1 Estimated output equations from FRM models

Consider the following complete set of fuzzy logic rules for the identified MISO fuzzy system

$$R_j^l:\ IF\ (x_1\ is\ X_1^{p_1})\ AND \cdots AND\ (x_n\ is\ X_n^{p_n}),\ THEN\ (y_j\ is\ Y_j^l),\ l=1,2,\cdots,M, \tag{4.12}$$

where $x=(x_1,x_2,\cdots,x_n)$, x_i is the i-th input variable on the universe of discourse E_i; $X_i^1,\cdots,X_i^{N_i}$ are fuzzy sets defined on E_i, $X_i^{p_i}\subset E_i$; N_i is the number of fuzzy sets, $i=1,2,\cdots,n$, $p_i\in\{1,2,\cdots,N_i\}$, $l=p_1+\sum_{i=2}^{n}[(p_i-1)\prod_{j=1}^{i-1}N_j]$; $M=\prod_{j=1}^{i-1}N_j$ is the total number of rules; y_j is the j-th output variable on the universe of discourse F_j; Y_j^l are fuzzy sets defined on F_j, $Y_j^l\subset F_j$, $l=1,2,\cdots,M$, $j=1,2,\cdots,m$.

Suppose that in each fuzzy rule Eq. (4.12), MFs of the output $\mu_{Y_j^l}(y_j)$, $l=1,2,\cdots,M$, are symmetrical and normal; in addition, each fuzzy set Y_j^l, parameters c_j^l are the MF centers of y_j. The center points of the fuzzy sets Y_j^l for output variables in the FRM model can be represented as

$$C_j=(c_j^1,\cdots,c_j^M)^T. \tag{4.13}$$

Given an input value $x_0=(x_{10},\cdots,x_{n0})$ with fuzzy sets $E_{x_i}=\{X_i^1,\cdots,X_i^{N_i}\}$, $i=1,\cdots,n$, similar to the operation of fuzzification, the input vector can be described as

$$V_E(x_0)=V_E(x_{10},\cdots,x_{n0})=V_{E_1}(x_{10})\rhd V_{E_2}(x_{20})\rhd\cdots\rhd V_{E_n}(x_{n0}), \tag{4.14}$$

where $V_{E_i}(x_{i0})=(\mu_{X_i^1}(x_{i0}),\cdots,\mu_{X_i^{N_i}}(x_{i0}))^T$, $i=1,\cdots,n$.

Assume the FRM with parameters c_j^l, $l=1,\cdots,M$, can be estimated by

$$M_R(x_1,x_2,\cdots,x_n;y_j)=V_F(y)\rhd V_E^T(x_1,\cdots,x_n). \tag{4.15}$$

Then, the estimated output vector can be obtained from Eq. (4.15) and (4.14) such as:

$$V_F(\hat{y}_0)=V_{F_j}(\hat{y}_{j0})=M_R(x_1,x_2,\cdots,x_n;y_j)\rhd V_E(x_0)\in\Re^{M\times 1}. \tag{4.16}$$

According to the definition of $V_{F_j}(\hat{y}_{j0})$, Eq. (4.16) can also be expressed as

$$R_E(x_0)=V_{F_j}(\hat{y}_{j0})=(\mu_{Y_j^1}(\hat{y}_{j0}),\cdots,\mu_{Y_j^M}(\hat{y}_{j0}))^T=(\gamma_j^1,\gamma_j^2,\cdots,\gamma_j^M)^T,\ \gamma_j^l=\mu_{Y_j^l}(\hat{y}_{j0}), \tag{4.17}$$

where the vector V_{F_j} of the estimated output can be determined by the real-valued FRM M_R with the input $x_0=(x_{10},\cdots,x_{n0})$.

Based on Eq. (4.17), the output $\hat{y}_{j0}$ can be estimated by the FRM model in Eq. (4.15), and can be obtained through defuzzification.

Assume that each output set is a singleton. Since $Y_j^1, \cdots, Y_j^M$ are normalized fuzzy sets with the centers c_j^l, $l = 1, \cdots, M$, using the center-average defuzzifier and $V_{F_j}(\hat{y}_{j0})$ in Eq. (4.16) and (4.17), we can obtain the real value of $\hat{y}_{j_0}$:

$$\hat{y}_{j0} = \frac{R_E^T(x_0)C_j}{H_{1\times M} R_E(x_0)} = \frac{\sum_{l=1}^{M} \gamma_j^l c_j^l}{\sum_{l=1}^{M} \gamma_j^l} = \frac{\gamma_j^1 \cdot c_j^1 + \cdots + \gamma_j^M \cdot c_j^M}{\gamma_j^1 + \cdots + \gamma_j^M}, \tag{4.18}$$

where

$$H_{1\times M} = (1 \quad 1 \quad \cdots \quad 1) \in \Re^{1\times M}. \tag{4.19}$$

Denote $\overline{R}_{x_0} = \dfrac{R_E^T(x_0)}{H_{1\times M} R_E(x_0)} \in \Re^{1\times M}$ and $C_j = (c_j^1, \cdots, c_j^M)^T \in \Re^{M\times 1}$, then,

$$\hat{y}_{j0} = \overline{R}_{x_0} C_j, \tag{4.20}$$

where C_j is a constant vector and $\overline{R}_{x_0}$ is the function of input variable x_0.

It is seen from Eq. (4.20) that $\hat{y}_{j_0}$ can be estimated by the FRM M_R, input variable x_0, and the output parameters C_j.

Example 4.1:

If $n = 2$, $N_1 = 2$, $N_2 = 3$, $M = 6$, then the vector expression of x_i will be

$$V_{E_i}(x_i) = (\mu_{X_i^1}(x_i), \cdots, \mu_{X_i^{N_i}}(x_i))^T = \begin{cases} (\mu_{X_1^1}(x_1), \mu_{X_1^2}(x_1))^T & for \quad i = 1 \\ (\mu_{X_2^1}(x_2),\ \mu_{X_2^2}(x_2),\ \mu_{X_2^3}(x_2))^T & for \quad i = 2. \end{cases}$$

The input vector of $x = (x_1, \cdots, x_i, \cdots, x_n) = (x_1, x_2)$ can be determined by

$V_E(x_1, x_2) = V_{E_1}(x_1) \triangleright V_{E_2}(x_2)$

$$(\mu_{X_1^1}(x_1) \wedge \mu_{X_2^1}(x_2),\ \mu_{X_1^1}(x_1) \wedge \mu_{X_2^2}(x_2), \mu_{X_1^1}(x_1) \wedge \mu_{X_2^3}(x_2), \mu_{X_1^2}(x_1) \wedge \mu_{X_2^1}(x_2),\ \mu_{X_1^2}(x_1) \wedge \mu_{X_2^2}(x_2), \mu_{X_1^2}(x_1) \wedge \mu_{X_2^3}(x_2))^T$$
$$:= (\gamma_1^1 \wedge \gamma_2^1, \gamma_1^1 \wedge \gamma_2^2, \gamma_1^1 \wedge \gamma_2^3,\ \gamma_1^2 \wedge \gamma_2^1, \gamma_1^2 \wedge \gamma_2^2, \gamma_1^2 \wedge \gamma_2^3)^T .$$

The vector of y_j can be determined by

$$\begin{aligned} V_{F_j}(y_j) &= (\mu_{C_j^1}(y_j), \cdots, \mu_{C_j^M}(y_j))^T \\ &= (\mu_{C_j^1}(y_j),\ \mu_{C_j^2}(y_j), \cdots,\ \mu_{C_j^6}(y_j))^T \\ &:= (\gamma_j^1 \quad \gamma_j^2 \quad \gamma_j^3 \quad \gamma_j^4 \quad \gamma_j^5 \quad \gamma_j^6)^T . \end{aligned}$$

Then, the FRM of $M_R(x_1, \cdots, x_n; y_j) = V_{F_j}(y_j) \triangleright V_E^T(x_1, \cdots, x_n)$ for the rule base will be

$M_R(x_1, x_2; y_j) = V_{F_j}(y_j) \triangleright V_E^T(x_1, x_2) =$

$$\begin{bmatrix} \gamma_j^1 \wedge \gamma_1^1 \wedge \gamma_2^1 & \gamma_j^1 \wedge \gamma_1^1 \wedge \gamma_2^2 & \gamma_j^1 \wedge \gamma_1^1 \wedge \gamma_2^3 & \gamma_j^1 \wedge \gamma_1^2 \wedge \gamma_2^1 & \gamma_j^1 \wedge \gamma_1^2 \wedge \gamma_2^2 & \gamma_j^1 \wedge \gamma_1^2 \wedge \gamma_2^3 \\ \gamma_j^2 \wedge \gamma_1^1 \wedge \gamma_2^1 & \gamma_j^2 \wedge \gamma_1^1 \wedge \gamma_2^2 & \gamma_j^2 \wedge \gamma_1^1 \wedge \gamma_2^3 & \gamma_j^2 \wedge \gamma_1^2 \wedge \gamma_2^1 & \gamma_j^2 \wedge \gamma_1^2 \wedge \gamma_2^2 & \gamma_j^2 \wedge \gamma_1^2 \wedge \gamma_2^3 \\ \gamma_j^3 \wedge \gamma_1^1 \wedge \gamma_2^1 & \gamma_j^3 \wedge \gamma_1^1 \wedge \gamma_2^2 & \gamma_j^3 \wedge \gamma_1^1 \wedge \gamma_2^3 & \gamma_j^3 \wedge \gamma_1^2 \wedge \gamma_2^1 & \gamma_j^3 \wedge \gamma_1^2 \wedge \gamma_2^2 & \gamma_j^3 \wedge \gamma_1^2 \wedge \gamma_2^3 \\ \gamma_j^4 \wedge \gamma_1^1 \wedge \gamma_2^1 & \gamma_j^4 \wedge \gamma_1^1 \wedge \gamma_2^2 & \gamma_j^4 \wedge \gamma_1^1 \wedge \gamma_2^3 & \gamma_j^4 \wedge \gamma_1^2 \wedge \gamma_2^1 & \gamma_j^4 \wedge \gamma_1^2 \wedge \gamma_2^2 & \gamma_j^4 \wedge \gamma_1^2 \wedge \gamma_2^3 \\ \gamma_j^5 \wedge \gamma_1^1 \wedge \gamma_2^1 & \gamma_j^5 \wedge \gamma_1^1 \wedge \gamma_2^2 & \gamma_j^5 \wedge \gamma_1^1 \wedge \gamma_2^3 & \gamma_j^5 \wedge \gamma_1^2 \wedge \gamma_2^1 & \gamma_j^5 \wedge \gamma_1^2 \wedge \gamma_2^2 & \gamma_j^5 \wedge \gamma_1^2 \wedge \gamma_2^3 \\ \gamma_j^6 \wedge \gamma_1^1 \wedge \gamma_2^1 & \gamma_j^6 \wedge \gamma_1^1 \wedge \gamma_2^2 & \gamma_j^6 \wedge \gamma_1^1 \wedge \gamma_2^3 & \gamma_j^6 \wedge \gamma_1^2 \wedge \gamma_2^1 & \gamma_j^6 \wedge \gamma_1^2 \wedge \gamma_2^2 & \gamma_j^6 \wedge \gamma_1^2 \wedge \gamma_2^3 \end{bmatrix} \in \Re^{6\times 6} .$$

If the current input variable is (x_{10}, x_{20}), the vectors of x_{10} and x_{20} can be represented as $V_{E_1}(x_{10}) = (\mu_{X_1'^1}(x_{10}), \mu_{X_1'^2}(x_{10}))$ and $V_{E_2}(x_{20}) = (\mu_{X_2'^1}(x_{20}), \mu_{X_2'^2}(x_{20}), \mu_{X_2'^3}(x_{20}))$. Then, the vector of inputs $x_0 = (x_{10}, x_{20})$ can be represented as

$$V_E(x_{10},\ x_{20}) = V_{E_1}(x_{10}) \triangleright V_{E_2}(x_{20}) .$$

By the FRM, the vector of the estimated output $\hat{y}_{j0}$ in Eq. (4.16) can be represented by

$$R_E(x_0) = V_{F_j}(\hat{y}_{j0}) = R_E(x_{10}, x_{20}) = M_R(x_1, x_2; y_j) \triangleright V_E(x_{10}, x_{20}) ,$$

and

$$V_{F_j}(\hat{y}_{j0}) = (\mu_{Y_j^1}(\hat{y}_{j0}), \cdots, \mu_{Y_j^6}(\hat{y}_{j0}))^T \in \Re^{6\times 1} .$$

Assume that max S-norm, min T-norm, singleton fuzzifiers, and center-average defuzzifier are used. Furthermore, suppose that the membership functions of Y_j^l are normal and symmetrical with centers c^l, $l = 1, 2, \cdots, 6$. Then, the estimated value of y_{j0} by the FRM model will be

$$\hat{y}_{j0} = \frac{R_E^T(x_0)C_j}{H_{1\times 6}R_E(x_0)} = \frac{\sum_{l=1}^{6}\mu_{Y_j^l}(\hat{y}_{j0})c_j^l}{\sum_{l=1}^{6}\mu_{Y_j^l}(\hat{y}_{j0})} = \frac{\mu_{Y_j^1}(\hat{y}_{j0})\cdot c_j^1 + \cdots + \mu_{Y_j^6}(\hat{y}_{j0})c_j^6}{\mu_{Y_j^1}(\hat{y}_{j0}) + \cdots + \mu_{Y_j^6}(\hat{y}_{j0})},$$

$$H_{1\times M} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1) \in \Re^{1\times 6},\ C_j = (c_j^1, \cdots, c_j^6)^T \in \Re^{6\times 1}.$$

Let $\overline{R}_{x_0} = \dfrac{R_E^T(x_0)}{H_{1\times 6}R_E(x_0)} \in \Re^{1\times 6}$, then, $\hat{y}_{j0} = \overline{R}_{x_0}C_j$.

As an example, assume the FRM M_R is obtained as

$$M_R = \begin{bmatrix} 0.9 & 0.7 & 0.5 & 0.3 & 0.1 & 0 \\ 0 & 1 & 0.8 & 0.6 & 0.4 & 0.2 \\ 0.4 & 0.6 & 0.9 & 0.7 & 0.5 & 0.3 \\ 0.3 & 0.5 & 0.9 & 1 & 0.8 & 0.6 \\ 0.2 & 0.4 & 0.7 & 0.9 & 1 & 0.8 \\ 0.1 & 0.3 & 0.5 & 0.3 & 0.1 & 1 \end{bmatrix} \in \Re^{6\times 6}.$$

If the input vector of $x_0 = (x_{10}, x_{20})$ is represented as

$$V_E(x_{10}, x_{20}) = (0.1 \quad 0.2 \quad 0.3 \quad 0.4 \quad 0.5 \quad 0.6)^T,$$

then, through min-max implication, the output vector of $\hat{y}_{j0}$ in Eq. (4.16) will be

$$V_{F_j}(\hat{y}_{j0}) = M_R(x_1, x_2, \cdots, x_n; y_j) \triangleright V_E(x_{10}, \cdots, x_{n0})$$

$$= \begin{bmatrix} 0.9 & 0.7 & 0.5 & 0.3 & 0.1 & 0 \\ 0 & 1 & 0.8 & 0.6 & 0.4 & 0.2 \\ 0.4 & 0.6 & 0.9 & 0.7 & 0.5 & 0.3 \\ 0.3 & 0.5 & 0.9 & 1 & 0.8 & 0.6 \\ 0.2 & 0.4 & 0.7 & 0.9 & 1 & 0.8 \\ 0.1 & 0.3 & 0.5 & 0.3 & 0.1 & 1 \end{bmatrix} \triangleright \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.6 \end{pmatrix}$$

$$= (0.3 \quad 0.4 \quad 0.5 \quad 0.6 \quad 0.6 \quad 0.6)^T .$$

If $C_j = (c_j^1, \cdots, c_j^M)^T = (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5)^T$, where c_j^l is the center of Y_j^l, then the estimated value $\hat{y}_{j0}$ by the FRM will be

$$\hat{y}_{j0} = \frac{R_E^T(x_0)C_j}{H_{1\times 6}R_E(x_0)} = \frac{0.3\times 0 + 0.4\times 1 + 0.5\times 2 + 0.6\times 3 + 0.6\times 4 + 0.6\times 5}{0.3 + 0.4 + 0.5 + 0.6 + 0.6 + 0.6} = 2.867 .$$

4.3.2 Indirect modeling procedure for the FRMs

In Section 4.2, the direct modeling is discussed based on input-output training data under the condition that all parameters are known for the fuzzy system. If the parameters $C_j = (c_j^1, \cdots, c_j^M)^T$ are unknown, the FRM system will be identified using matrix models in Eq. (4.18)-(4.20) by using some appropriate optimization algorithm as discussed below.

Given Q pairs of input-output training data sets:

$$(x_{1q}, \cdots, x_{nq}, y_{jq}), \quad q = 1, \cdots, Q . \tag{4.21}$$

The FRM for inputs for each pair of data will be

$$V_E\ (x_{1q}, \cdots, x_{nq}) = V_{E_1}(x_{1q}) \triangleright V_{E_2}(x_{2q}) \triangleright \cdots \triangleright V_{E_n}(x_{nq}) , \tag{4.22}$$

where $V_{E_i}(x_{iq}) = (\mu_{X_i^1}(x_{iq}), \cdots, \mu_{X_i^{N_i}}(x_{iq}))^T$, $i = 1, \cdots, n$.

After fuzzy reasoning through the STP of logic matrices and defuzzification (e.g., center-average), the outputs can be computed from the FRM model in Eq. (4.18)-(4.20):

$$\hat{y}_{jq} = \overline{R}_{x_q} C_j , \ q = 1, \cdots, Q , \tag{4.23}$$

where $C_j = (c_j^1, \cdots, c_j^M)^T \in \Re^{M\times 1}$, $\overline{R}_{x_q} = \dfrac{R_E^T(x_q)}{H_{1\times M}R_E(x_q)} \in \Re^{1\times M}$, $H_{1\times M} = (1 \quad 1 \quad \cdots \quad 1) \in \Re^{1\times M}$.

The model outputs for all data in Eq. (4.21) will be

$$\hat{Y}_j = \overline{R}_x C_j , \tag{4.24}$$

where $\hat{Y}_j = (\hat{y}_{j1}, \hat{y}_{j2}, \cdots, \hat{y}_{jQ})^T \in \Re^{Q\times 1}$, $\overline{R}_x = (\overline{R}_{x1}^T, \overline{R}_{x2}^T, \cdots, \overline{R}_{xQ}^T)^T \in \Re^{Q\times M}$; Q is much larger than M for FRM modeling adequacy.

In general, it is difficult to get exact solutions of Eq. (4.24). Therefore, we will search for $C_j = \hat{C}_j$ by minimizing the following error function

$$E(C_j) = \sum_{q=1}^{Q} (\overline{R}_{x_q} C_j - y_{jq})^2 = e^T e = (\overline{R}_{x_q} C_j - Y_j)^T (\overline{R}_{x_q} C_j - Y_j), \tag{4.25}$$

where $e = \overline{R}_{x_q} C_j - Y_j$ is the error vector, and Y_j is defined by

$$Y_j = (y_{j1}, y_{j2}, \cdots, y_{jQ})^T. \tag{4.26}$$

The function $E(C_j)$ in Eq. (4.25) is in a quadratic form, which can be minimized at $C_j = \hat{C}_j$ by the LSE [149]:

$$\min_{C_j} Y_j = \left\| \overline{R}_{x_q} C_j - Y_j \right\|_2. \tag{4.27}$$

A necessary condition to get $\hat{C}_j$ in Eq. (4.27) is to satisfy the following normal condition

$$\overline{R}^T_{x_q} \overline{R}_{x_q} \hat{C}_j = \overline{R}^T_{x_q} Y_j. \tag{4.28}$$

If $\overline{R}^T_{x_q} \overline{R}_{x_q}$ is nonsingular, $\hat{C}_j$ is unique and can be computed as

$$\hat{C}_j = (\overline{R}^T_{x_q} \overline{R}_{x_q})^{-1} \overline{R}^T_{x_q} Y_j. \tag{4.29}$$

The proof of Eq. (4.29) can be seen in Appendix A.

The above LSE can be generalized to MIMO fuzzy systems with m outputs. If the output variables are independent of each other, and for y_j, the center points $C_j = (c_j^1, \cdots, c_j^M)^T$ of fuzzy sets $Y_j^1, \cdots, Y_j^M$, $j = 1, \cdots, m$, can be determined by the LSE similar to the indirect modeling of the MISO FRM in Eq. (4.21) to (4.29).

4.4 Numerical Simulation Examples

The effectiveness of the proposed direct and indirect FRM modeling techniques will be examined using some simulation examples.

Example 4.2:

Suppose a fuzzy system has two inputs (x_1, x_2), and one output $y_1 = x_1 + x_2$. The universe of discourse for x_i is [0, 2], $i = 1,\ 2$; and the universe of discourse for y_1 is [0, 4]. Three MFs for fuzzy sets { X_i^1, X_i^2, X_i^3 } are assigned for each input x_i as illustrated in Fig. 4.1, and nine triangular MFs for fuzzy sets { Y_1^1, Y_1^2, ..., Y_1^9 } are assigned for the output variable y_1 as shown in Fig. 4.2. AND and OR operators are chosen as min and max operations. The fuzzy rules will be formulated as follows:

$$R^1: IF\ (x_1\ is\ X_1^1)\ and\ (x_2\ is\ X_2^1),\ THEN\ (y_1\ is\ Y_1^1);$$

$$R^2: IF\ (x_1\ is\ X_1^1)\ and\ (x_2\ is\ X_2^2),\ THEN\ (y_1\ is\ Y_1^2);$$

$$\vdots$$

$$R^9: IF\ (x_1\ is\ X_1^3)\ and\ (x_2\ is\ X_2^3),\ THEN\ (y_1\ is\ Y_1^9).$$

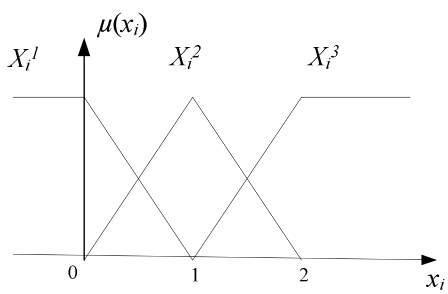


Fig. 4.1. The fuzzy sets with membership functions of input variables.

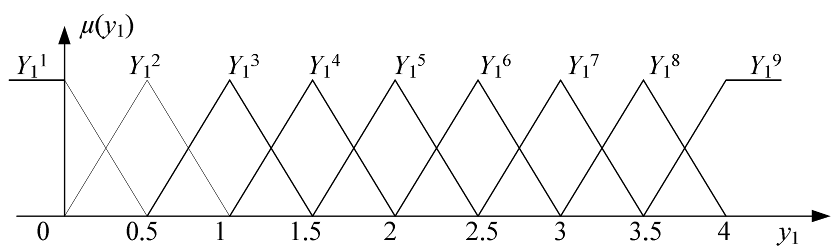


Fig. 4.2. The fuzzy sets with membership functions of the output variable.

Consider P = 121 training data pairs $(x_{1p}^*,\ x_{2p}^*;\ \ y_{1p}^*)$, $p = 1,\ 2, \cdots,\ 121$. The FRM model will be identified for the fuzzy relation between the input and the output. For example, if $(x^*_{1p}, x^*_{2p};\ y^*_{1p}) = (0.2,\ 1.3;\ 1.5)$, the following vectors can be determined according to MFs in Fig. 4.1 and 4.2:

$$V_{E_1}(x^*_{1p}) = (\mu_{X_1^1}(x^*_{1p}),\ \mu_{X_1^2}(x^*_{1p}),\ \mu_{X_1^3}(x^*_{1p}))^T = (0.8,\ 0.2,\ 0)^T,$$

$$V_{E_2}(x^*_{2p}) = (\mu_{X_2^1}(x^*_{2p}),\ \mu_{X_2^2}(x^*_{2p}),\ \mu_{X_2^3}(x^*_{2p}))^T = (0,\ 0.7,\ 0.3)^T,$$

$$V_{F_1}(y^*_{1p}) = (\mu_{Y_1^1}(y^*_{1p}),\ \mu_{Y_1^2}(y^*_{1p}),\ \ldots, \mu_{Y_1^9}(y^*_{1p}))^T = (0, 0, 0, 1, 0, 0, 0, 0, 0)^T.$$

The fuzzy relation vector of the inputs from Eq. (4.7) will be

$$V_E\ (x^*_{1p}, x^*_{2p}) = V_{E_1}(x^*_{1p}) \triangleright V_{E_2}(x^*_{2p}) = (0, 0.7, 0.3, 0, 0.2, 0.2, 0, 0, 0)^T.$$

The fuzzy relation vector of the output from Eq. (4.8) will be

$$V_{F_1}(y^*_{1p}) = (0, 0, 0, 1, 0, 0, 0, 0, 0)^T.$$

The i-th FRM model between inputs and the output from Eq. (4.9) will be

$$M_{R_{1p}}\left(x^*_{1p}, x^*_{2p};\ y^*_{1p}\right) := V_E(x^*_{1p}, x^*_{2p}) \triangleright V^T_{F_1}(y^*_{1p}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.7 & 0.3 & 0 & 0.2 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

If $\bigcup$ is chosen as a "max" operator in Eq. (4.10), the resulting FRM model from these 121 training data pairs will be

$$M_{R_1}\left(x_1, x_2;\ y_1\right) := \bigcup_{p=1}^{121} M_{R_{1p}}\left(x^*_{1p}, x^*_{2p};\ y^*_{1p}\right)$$

$$
= \begin{pmatrix}
0.83 & 0.57 & 0.16 & 0.56 & 0.55 & 0.196 & 0.19 & 0.19 & 0.10 \\
0.87 & 0.77 & 0.35 & 0.78 & 0.66 & 0.35 & 0.38 & 0.38 & 0.19 \\
0.75 & 0.97 & 0.57 & 0.99 & 0.78 & 0.55 & 0.55 & 0.55 & 0.31 \\
0.65 & 0.87 & 0.76 & 0.87 & 0.88 & 0.65 & 0.78 & 0.66 & 0.42 \\
0.53 & 0.75 & 0.96 & 0.76 & 0.98 & 0.78 & 0.95 & 0.77 & 0.55 \\
0.43 & 0.63 & 0.72 & 0.65 & 0.87 & 0.88 & 0.78 & 0.86 & 0.65 \\
0.32 & 0.54 & 0.56 & 0.53 & 0.76 & 0.95 & 0.53 & 0.97 & 0.77 \\
0.21 & 0.37 & 0.37 & 0.35 & 0.67 & 0.76 & 0.35 & 0.75 & 0.85 \\
0.09 & 0.15 & 0.15 & 0.17 & 0.54 & 0.57 & 0.17 & 0.57 & 0.72
\end{pmatrix}.
$$

Consider an input vector $(x_{10}, x_{20}) = (0.8 \ \ 1.35)$, the vector of the output variable y_{10} will be

$$
V_F(\hat{y}) = V_{F_1}(\hat{y}_{10}) = M_{R_1} \rhd V_E(x_{10}, x_{20}) = M_{R_1} \rhd (V_{E_1}(x_{10}) \rhd V_{E_2}(x_{20}))
$$

$$
= (0.325, 0.6, 0.6, 0.325, 0.825, 0.675, 0.325, 0.4, 0.4)^T.
$$

With the center of MFs { Y_1^1 , Y_1^2 , ..., Y_1^9 }, $C_1 = (c_1^1, c_1^2, \cdots, c_1^9)^T$ $= (0 \ \ 0.5 \ \ 1 \ \ 1.5 \ \ 2 \ \ 2.5 \ \ 3 \ \ 3.5 \ \ 4)^T$, by center-average defuzzification, the output can be calculated by

$$
\hat{y}_{10} = \frac{R_E^T(x_0)C_j}{H_{1\times 9}R_E(x_0)} = \frac{V_{F_1}(y_{10})C_1}{\sum_{j=1}^{9}\mu_{Y_1^j}(y_{10})} = \frac{\sum_{j=1}^{9}\left(c_1^j \mu_{Y_1^j}(y_{10})\right)}{\sum_{j=1}^{9}\mu_{Y_1^j}(y_{10})}
$$

$$
= \frac{0\times 0.325 + 0.5\times 0.6 + 1\times 0.6 + 1.5\times 0.325 + 2\times 0.825 + 2.5\times 0.675 + 3\times 0.325 + 3.5\times 0.4 + 4\times 0.4}{0.325 + 0.6 + 0.6 + 0.325 + 0.825 + 0.675 + 0.325 + 0.4 + 0.4}
$$

$$
= \frac{8.7}{4.475} = 1.944\,. \tag{4.30}
$$

Correspondingly, the error between the actual output and the output from the fuzzy model is 6.05%. If each input variable has 11 fuzzy sets, the output of the fuzzy model in Eq. (4.30) will be $y_{10} = 2.150$ with 0.01% error. Based on more simulation tests with different number of fuzzy sets for input variables, for the fixed 121 training data pairs as the modeling data base, when the same universe of discourse and similar MF parameters are chosen, 11 is the optimal number of input fuzzy sets in this fuzzy system.

Example 4.3:

For Example 4.2, if the center points of MFs for outputs are unknown, the indirect FRM method can be used for system modeling, using Eq. (4.22) to Eq. (4.29).

Suppose the centers of the fuzzy sets for the output are $(c_{1,1}, c_{1,2}, \cdots, c_{1,9})$. If the training data pairs are $(x^*_{1p}, x^*_{2p};\ y^*_{1p}),\ \ p = 1, 2, \cdots, 121$, then

$$V_E\,(x^*_{1p}, x^*_{2p}) = V_{E_1}(x^*_{1p}) \triangleright V_{E_2}(x^*_{2p}),\ \ p = 1, 2, \cdots, 121.$$

$Y = \overline{R}_{x_p} C_1 + e_1$ can be expressed as

$$\begin{pmatrix} Y^*_{1,1} \\ Y^*_{1,2} \\ \vdots \\ Y^*_{1,121} \end{pmatrix} = \begin{pmatrix} R^T_{x_1}(x^*_{1,1}, x^*_{2,1}) \\ R^T_{x_2}(x^*_{1,2}, x^*_{2,2}) \\ \vdots \\ R^T_{x_{121}}(x^*_{1,121}, x^*_{2,121}) \end{pmatrix} \begin{pmatrix} c^*_{1,1} \\ c^*_{1,2} \\ \vdots \\ c^*_{1,9} \end{pmatrix} + e_1 .$$

Using the LSE, the estimated outputs will be

$$\begin{pmatrix} c_{1,1} \\ c_{1,2} \\ \vdots \\ c_{1,9} \end{pmatrix} = (\overline{R}^T_x \overline{R}_x)^{-1} \overline{R}^T_x Y = \begin{pmatrix} -0.0221 \\ 0.9822 \\ 2.0000 \\ 0.9822 \\ 2.0000 \\ 3.0178 \\ 2.0000 \\ 3.0178 \\ 4.0221 \end{pmatrix}, \text{ where}$$

$$Y = \begin{pmatrix} Y^*_{1,1} \\ Y^*_{1,2} \\ \vdots \\ Y^*_{1,121} \end{pmatrix},\ \overline{R}_x = \begin{pmatrix} R^T_{x_1}(x^*_{1,1},\ x^*_{2,1}) \\ R^T_{x_2}(x^*_{1,2},\ x^*_{2,2}) \\ \vdots \\ R^T_{x_{121}}(x^*_{1,121},\ x^*_{2,121}) \end{pmatrix}.$$

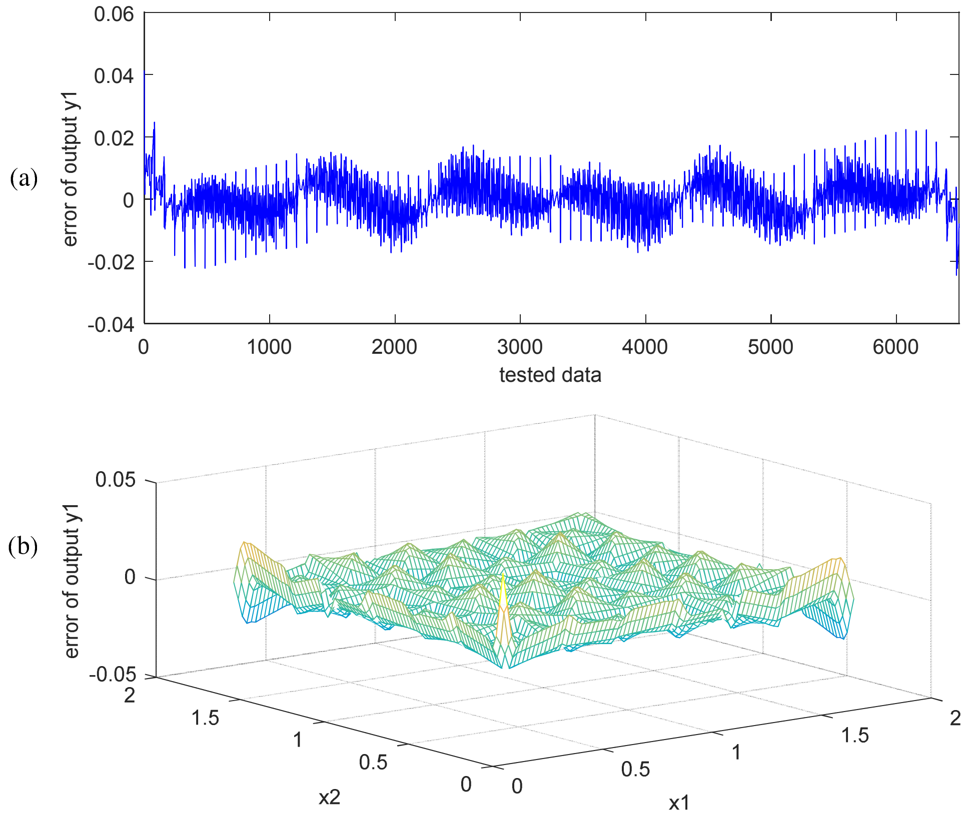


Fig. 4.3. Errors between real outputs and FRM model outputs: (a) Errors with respect to tested data; (b) Errors with respect to the inputs.

With these estimated centers values of output variables, the output y_1 from the FRM model can be computed by Eq. (4.29). The errors are illustrated in Fig. 4.3, with an average error 0.37%. The relationship between input and output variables is illustrated in Fig. 4.4.

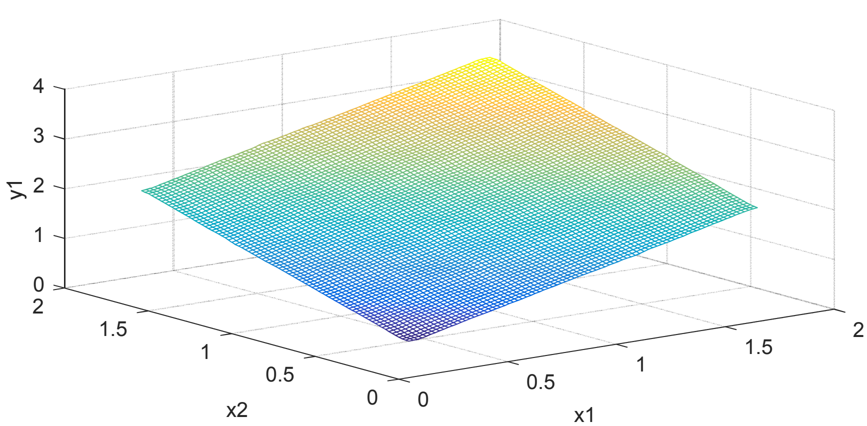


Fig. 4.4. The relationship between input and output variables.

Moreover, as the fuzzy systems are generated from the same function $y=x_1+x_2$, Comparison results can be obtained between the average error from the indirect modeling method and the errors obtained by the direct modeling method in Example 4.2. It is obvious that the simulation results by the indirect modeling method are much better than the results in direct modeling method by FRM models.

Example 4.4:

Consider Example 4.3. Assume the fuzzy system has two output variables generated by $y_1 = x_1 + x_2$, $y_2 = x_1 x_2$ and the center points of MFs for output variables are unknown. The FRM model can be determined using the indirect modeling methods in Eq. (4.21-4.29).

Suppose the centers of y_1 and y_2 are $(c_{1,1}, c_{1,2}, \cdots, c_{1,9})$ and $(c_{2,1}, c_{2,2}, \cdots, c_{2,9})$, respectively. If 10000 input-output data pairs are selected as $(x^*_{1q}, x^*_{2q};\ y^*_{1q}, y^*_{2q})$, $q = 1, 2, \cdots, 10000$, the FRM for two input variables can be determined as $V_E(x^*_{1q}, x^*_{2q}) = V_{E_1}(x^*_{1q}) \triangleright V_{E_2}(x^*_{2q})$, $q = 1, 2, \cdots, 10000$.

For a two-input two-output (TITO) fuzzy system, $Y_j = \overline{R}_{x_q} C_j + e$, $j = 1, 2$, can be represented as

$$Y_1 = \begin{pmatrix} y^*_{1,1} \\ y^*_{1,2} \\ \vdots \\ y^*_{1,10000} \end{pmatrix} = \begin{pmatrix} R^T_{x1}(x^*_{1,1}, x^*_{2,1}) \\ R^T_{x2}(x^*_{1,2}, x^*_{2,2}) \\ \vdots \\ R^T_{x10000}(x^*_{1,121}, x^*_{2,10000}) \end{pmatrix} \begin{pmatrix} c^*_{1,1} \\ c^*_{1,2} \\ \vdots \\ c^*_{1,9} \end{pmatrix} + e_1, \text{ and}$$

$$Y_2 = \begin{pmatrix} y^*_{2,1} \\ y^*_{2,2} \\ \vdots \\ y^*_{2,10000} \end{pmatrix} = \begin{pmatrix} R^T_{x1}(x^*_{1,1}, x^*_{2,1}) \\ R^T_{x2}(x^*_{1,2}, x^*_{2,2}) \\ \vdots \\ R^T_{x10000}(x^*_{1,121}, x^*_{2,10000}) \end{pmatrix} \begin{pmatrix} c^*_{2,1} \\ c^*_{2,2} \\ \vdots \\ c^*_{2,9} \end{pmatrix} + e_2.$$

Similar to the calculation in Example 4.3, based on the LSE, the estimated values of $Y = \overline{R}_x C + e$ will be

$$\hat{C}_1 = \begin{pmatrix} c^*_{1,1} \\ c^*_{1,2} \\ \vdots \\ c^*_{1,9} \end{pmatrix} = (\overline{R}_x^T \overline{R}_x)^{-1} \overline{R}_x^T Y_1 = \begin{pmatrix} -1.601 \\ -1.183 \\ 1.761 \\ 0.540 \\ 2.669 \\ 5.476 \\ 1.935 \\ 2.521 \\ 4.846 \end{pmatrix} \text{ and } \hat{C}_2 = \begin{pmatrix} c^*_{2,1} \\ c^*_{2,2} \\ \vdots \\ c^*_{2,9} \end{pmatrix} = (\overline{R}_x^T \overline{R}_x)^{-1} \overline{R}_x^T Y_2 = \begin{pmatrix} 0.159 \\ -1.144 \\ -2.204 \\ -1.087 \\ 1.005 \\ 1.982 \\ -1.708 \\ 3.701 \\ 6.488 \end{pmatrix}.$$

Using the calculated $\hat{C}_1$ and $\hat{C}_2$, the output $y = (y_1, y_2)$ of Eq. (4.23) can be determined from the FRM model. The mean of the errors between the actual outputs and modeling outputs is 0.0094 for y_1 and 0.0137 for y_2, respectively, as illustrated in Fig. 4.5.

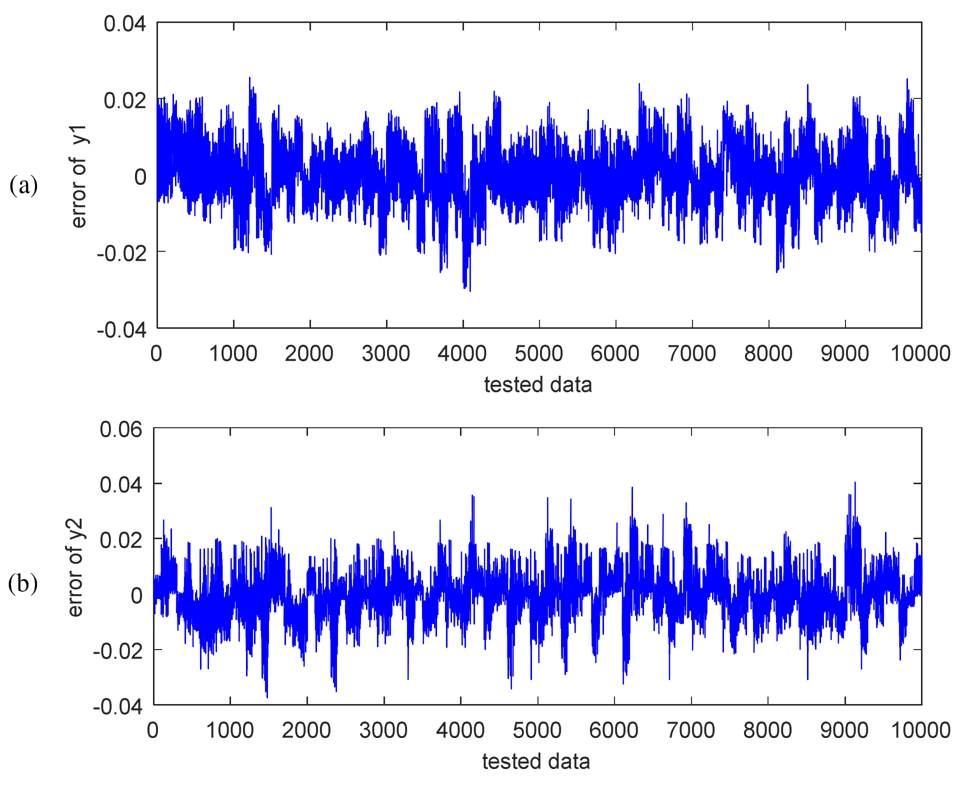


Fig. 4.5. Errors between real outputs and fuzzy model outputs: (a) Errors of y_1 for tested data; (b) Errors of y_2 for tested data.

Based on these simulation examples, it is seen that the proposed direct and indirect modeling techniques are efficient in formulating FRM models for fuzzy systems. They are feasible for MIMO fuzzy system analysis.

4.5 Concluding Remarks

In this chapter, two new FRM modeling techniques have been proposed to transform the fuzzy logic reasoning based on fuzzy rules into a problem of solving algebraic equations by fuzzy STP operations. It can be formulated using a direct modeling method with known parameters for all fuzzy variables or an indirect identification method for fuzzy systems with unknown parameters in output variables. The effectiveness of the proposed FRM modeling methods is verified by a series of simulation examples. The related contribution in this chapter has been published (or accepted for publication) in:

(1) H. Lyu, W. Wang, X. Liu, "Modeling of multi-variable fuzzy systems by semi-tensor product", *IEEE Transactions on Fuzzy Systems*, Vol. 28, No. 2, pp. 228-235, Feb. 2020.

Chapter 5 Universal Approximation of FRM Models

In fuzzy model approximation, the current literature mainly focused on universal approximation of MISO Takagi-Sugeno-Kang (TSK) fuzzy systems [150, 151], but not on general multi-variable fuzzy systems due to the deficiency of proper mathematical tools [152, 153]. The objective of this chapter is to extend the universal approximation of fuzzy models to general MIMO FRM models based on the STP. It is novel in the following aspects: 1) An FRM formulation technique is proposed to approximate arbitrary nonlinear functions by MIMO fuzzy systems. 2) The fuzzy reasoning operations can be realized by the fuzzy logic STP in universal approximation of the FRM model.

The rest of this chapter is organized as follows. Firstly, some related FRM preliminaries are introduced in Section 5.1. Section 5.2 discusses the universal approximation of FRM systems. The design algorithms of FRM models are discussed in Section 5.3. The approximation accuracy of FRM models is investigated in Section 5.4; and the effectiveness of the proposed universal approximation technique is tested by some simulations in Section 5.5.

5.1 Some Related Preliminaries

In this chapter, the universal approximation of FRM models is extended to multi-variable fuzzy systems, especially nonlinear functions. Consider a general FRM system with *n*-inputs and *m*-outputs, $x=(x_1, x_2, \cdots, x_n)$ and $y=(y_1, y_2, \cdots, y_m)$. The corresponding fuzzy rule structure will be

$$R^l: \ IF \ (x_1 \ is \ X_1^{p_1}) \ AND \cdots AND \ (x_n \ is \ X_n^{p_n}),$$

$$THEN \ (y_1 \ is \ Y_1^l) \ AND \cdots AND \ (y_m \ is \ Y_m^l), \ l=1,2,\cdots,M, \qquad (5.1)$$

where $M = N_1 N_2 \cdots N_n$ is the number of fuzzy rules; N_i is the number of fuzzy sets, $i = 1,2,\cdots,n$; x_i is the i-th input variable on the universe of discourse E_i; $X_i^1,\cdots, X_i^{N_i}$ are fuzzy sets defined on E_i, $p_i \in \{1, 2, \cdots, N_i\}$, $l = p_1 + \sum_{i=2}^{n}[(p_i - 1)\prod_{j=1}^{i-1} N_j]$; y_k is the k-th output variable on the universe of discourse F_k; $Y_k^1,\cdots, Y_k^M$ are fuzzy sets defined on F_k, $k = 1,\ 2,\cdots, m$.

If all of the outputs are independent with each other, Eq. (5.1) is equivalent to m MISO fuzzy systems. Without loss of generality, consider one MISO FRM with the following rule set

$$R_k^l : IF\ (x_1\ is\ X_1^{p_1})\ AND \cdots AND\ (x_n\ is\ X_n^{p_n}),\ THEN\ (y_k\ is\ Y_k^l),$$

$$l = 1, 2, \cdots, M,\ k = 1, 2, \cdots, m. \quad (5.2)$$

1) Based on the FRM modeling in Chapter 4, the input vector is modeled by $V_E(x_1,\cdots, x_n) = V_{E_1}(x_1) \triangleright V_{E_2}(x_2) \triangleright \cdots \triangleright V_{E_n}(x_n)$, $V_{E_i}(x_i) = (\mu_{X_i^1}(x_i),\cdots, \mu_{X_i^{N_i}}(x_i))^T$.

2) Assume that in each fuzzy rule Eq. (5.2), MFs of the output $\mu_{Y_k^l}(y_k)$ are symmetrical and normal (i.e., convex fuzzy sets and MF grade $\mu_{Y_k^l}(y_k) = 1$, $y_k \in Y_k^l$). Suppose that on each fuzzy set Y_k^l, c_k^l is the center of all of values for y_k, $l = 1, 2, \cdots, M$. Then, the output vector model can be determined by the output vector $V_{F_k}(y_k) = (\mu_{Y_k^1}(y_k),\cdots, \mu_{Y_k^M}(y_k))^T$.

3) Assume that the real-valued FRM model $M_{R_k} := M_{R_k}(x_1, x_2, \cdots, x_n; y_k)$ has been constructed using Eq. (4.1) to Eq. (4.9). Given an input $x(t) = (x_1(t), x_2(t), \cdots, x_n(t))$, the vector expression for the corresponding output $y_k(t)$ from the FRM model M_{R_k} can be calculated by Eq. (4.16):

$$R_E(x(t), y_k(t)) = M_{R_k} \triangleright V_E(x_1(t), x_2(t), \cdots, x_n(t)). \quad (5.3)$$

4) Use an appropriation composition operator (e.g., max-min, max-product), a singleton output, and the center-average defuzzifier in Eq. (5.2). If $C_k = (c_k^1,\cdots, c_k^M)^T$,

according to $R_E = V_{F_k} = (\mu_{Y_k^1}(y_k(t)), \cdots, \mu_{Y_k^M}(y_k(t)))^T$ and Eq. (4.18), the estimated output $y_k(t)$ of the FRM for the MISO fuzzy system in Eq. (5.2) can be determined by

$$y_k(t) := f_{FRM}(x_1(t), x_2(t), \cdots, x_n(t)) = \overline{R}_{x_k}(t)C_k, \tag{5.4}$$

where

$$C_k = (c_k^1, \cdots, c_k^M)^T,\ \overline{R}_{x_k}(t) = \frac{R_E^T(x(t), y_k(t))}{H_{1\times M} R_E(x(t), y_k(t))},\ H_{1\times M} = (1 \quad 1 \quad \cdots \quad 1) \in \Re^{1\times M}. \tag{5.5}$$

It is seen from Eq. (5.4) that a fuzzy system in Eq. (5.2) with an FRM model is a nonlinear mapping from $x = (x_1, \cdots, x_n) \in \Re^n$ to $f_{FRM}(x) \in \Re$, represented by the product of two matrices. Fig. 5.1 demonstrates fuzzy logic reasoning procedures of a general MISO FRM model of a fuzzy system by the nonlinear mapping $f_{FRM}(x_1, \cdots, x_n)$ in Eq. (5.4).

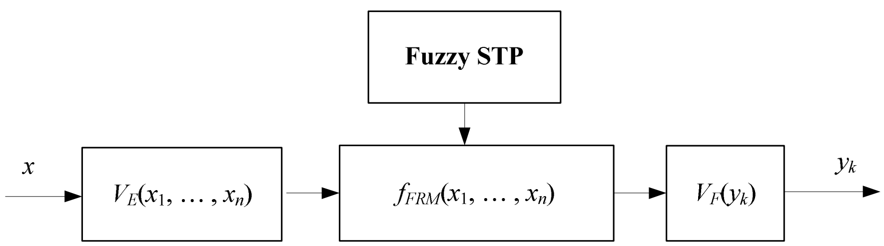


Fig. 5.1. The structure of the FRM system.

On the other hand, the MISO FRM model in Eq. (5.2) can be extended to an MIMO FRM model in Eq. (5.1). Suppose that the output fuzzy sets Y_k^l are symmetrical and normal with centers c_k^l, $l = 1, \cdots, M$, $k = 1, 2, \cdots m$. Using the above procedure from 1) to 4) and the rule base in Eq. (5.1), if a max-product reasoning operator and a center-average defuzzifier are used, the output becomes:

$$\begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{pmatrix} = \begin{pmatrix} f_{FRM}^1(x(t)) \\ f_{FRM}^2(x(t)) \\ \vdots \\ f_{FRM}^m(x(t)) \end{pmatrix} = \begin{pmatrix} \overline{R}_{x_1}(t)C_1 \\ \overline{R}_{x_2}(t)C_2 \\ \vdots \\ \overline{R}_{x_m}(t)C_m \end{pmatrix}$$

$$=\begin{pmatrix} \overline{R}_{x_1}(t) & 0 & 0 & \cdots & 0 \\ 0 & \overline{R}_{x_2}(t) & 0 & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & 0 & 0 & \overline{R}_{x_m}(t) \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{pmatrix} = \overline{R}_x(t)C, \tag{5.6}$$

where $x(t) = (x_1(t), x_2(t), \cdots, x_n(t))$, $C_k = (c_k^1, \cdots, c_k^M)^T \in \Re^{M\times 1}$,

$\overline{R}_{x_k}(t) = \dfrac{R_E^T(x(t))}{H_{1\times M} R_E(x(t))} \in \Re^{1\times M}$, $H_{1\times M} = (1 \quad 1 \quad \cdots \quad 1) \in \Re^{1\times M}$,

$y(t) = (y_1(t), y_2(t), \cdots, y_m(t))^T \in \Re^{m\times 1}$, $k = 1, 2, \cdots, m$,

$$\overline{R}_x(t) = \begin{pmatrix} \overline{R}_{x_1}(t) & 0 & 0 & \cdots & 0 \\ 0 & \overline{R}_{x_2}(t) & 0 & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & 0 & 0 & \overline{R}_{x_m}(t) \end{pmatrix} \in \Re^{m\times(mM)},\ C = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{pmatrix} \in \Re^{mM\times 1}. \tag{5.7}$$

The difference between Eq. (5.4) and Eq. (5.6) is that the fuzzy relation function of MISO systems is a scalar whereas the fuzzy relation function of MIMO systems is a column vector, or $\overline{R}_{x_k}(t)$ vs. $\overline{R}_x(t)$, and $C_k = (c_{k1}, \cdots, c_{kM})^T$ vs. $C = (C_1^T, C_2^T, \cdots, C_m^T)^T$.

5.2 Universal Approximation by FRM Models

The following theorem could be used to approximate a continuous multi-variable function by a MISO FRM model.

Theorem 5.1 (**Universal approximation by a MISO FRM**): Suppose that $\tilde{E} = \prod_{i=1}^{n} E_i$, $E_i \subset \Re$, is a complete set on $\Re^n$. For a given real-valued continuous function $g(x)$ on $\tilde{E}$ with $x = (x_1, x_2, \cdots, x_n) \in \tilde{E}$ and an arbitrary real number $\varepsilon > 0$, there exists an FRM model $f_{FRM}(x)$ in the form of Eq. (5.4), such that

$$\sup_{x\in\tilde{E}} \left| f_{FRM}(x) - g(x) \right| < \varepsilon. \tag{5.8}$$

Such an FRM model $f_{FRM}(x)$ is a universal approximator.

Proof: The proof is based on the Stone-Weierstrass Theorem [32]. If $F_{FRM}(\tilde{E})$ is a set of fuzzy systems $f_{FRM}(x)$ in the form of Eq. (5.4), it is to verify that $F_{FRM}(\tilde{E})$ is an algebra; $F_{FRM}(\tilde{E})$ can separate points on $\tilde{E}$; and $F_{FRM}(\tilde{E})$ vanishes at no point of $\tilde{E}$.

1) To verify that $F_{FRM}(\tilde{E})$ is an algebra, i.e., $F_{FRM}(\tilde{E})$ is closed under addition, multiplication and scalar multiplication.

Given $f^1_{FRM}(x)$, $f^2_{FRM}(x) \in F_{FRM}(\tilde{E})$, then

$$f^1_{FRM}(x) + f^2_{FRM}(x) = \overline{R}^1_{x_k} C^1_k + \overline{R}^2_{x_k} C^2_k = (\overline{R}^1_{x_k} \quad \overline{R}^2_{x_k})((C^1_k)^T \quad (C^2_k)^T)^T = \overline{R}^*_{x_k} C^*_k \in F_{FRM}(\tilde{E}),$$

where $\overline{R}^*_{x_k} = (\overline{R}^1_{x_k} \quad \overline{R}^2_{x_k})$, $C^*_k = ((C^1_k)^T \quad (C^2_k)^T)^T$.

Similarly, $f^1_{FRM}(x) \cdot f^2_{FRM}(x) = \overline{R}^1_{x_k} C^1_k \cdot \overline{R}^2_{x_k} C^2_k = (\overline{R}^1_{x_k} C^1_k \overline{R}^2_{x_k}) C^2_k \in F_{FRM}(\tilde{E})$.

Given a $c \in \Re$, $cf^1_{FRM}(x) = c\overline{R}^1_{x_k} C^1_k = (c\overline{R}^1_{x_k}) C^1_k \in F_{FRM}(\tilde{E})$.

Hence, $F_{FRM}(\tilde{E})$ is an algebra.

2) To verify that $F_{FRM}(\tilde{E})$ can separate points on $\tilde{E}$, i.e., for arbitrary $x^1 = (x^1_1, \cdots, x^1_n)$ and $x^2 = (x^2_1, \cdots, x^2_n) \in \tilde{E}$; if $x^1 \neq x^2$, then, there exists an $f_{FRM}(x) \in F_{FRM}(\tilde{E})$ such that $f_{FRM}(x_1) \neq f_{FRM}(x_2)$.

Choose parameters of $f_{FRM}(x)$ in the form of Eq. (5.4). If $x^1 \neq x^2$, there exists at least one i: $1 \le i \le n$ and one fuzzy set $X^{j_1}_i$, such that $\mu_{X^{j_1}_i}(x^1_i) \neq \mu_{X^{j_1}_i}(x^2_i)$, or $V_{E_i}(x^1_i) \neq V_{E_i}(x^2_i)$. According to Eq. (4.16)-(4.20), there exist $V_E(x^1) = V_{E_1}(x^1_1) \rhd V_{E_2}(x^1_2) \rhd \cdots \rhd V_{E_n}(x^1_n)$ and $V_E(x^2) = V_{E_1}(x^2_1) \rhd V_{E_2}(x^2_2) \rhd \cdots \rhd V_{E_n}(x^2_n)$, then it can verify that $\overline{R}_{x^1_k} \neq \overline{R}_{x^2_k}$, $\overline{R}_{x^1_k} C_k \neq \overline{R}_{x^2_k} C_k$. Thus, $F_{FRM}(\tilde{E})$ can separate points on $\tilde{E}$ by the fuzzy system $f_{FRM}(x)$.

3) To verify that $F_{FRM}(\tilde{E})$ vanishes at no point of $\tilde{E}$, i.e., for each $x = (x_1, x_2, \cdots, x_n) \in \tilde{E}$, there exists an $f_{FRM}(x) \in F_{FRM}(\tilde{E})$ such that $f_{FRM}(x) \neq 0$.

If $\mu_{X_i^l}(x_i) \equiv 0$, $i = 1, 2, \cdots, n$, $l = 1, 2, \cdots, N_i$, for an arbitrary x_i, $\overline{R}_{y_k}$ is zero. However, if $\mu_{X_i^l}(x_i) \equiv 0$, then, $R_E(x) = V_{F_k}(y_k(t) = 0$ because of $R_E = M_R \triangleright V_E$. Consequently, the denominator of $\overline{R}_{x_k}(t) = \dfrac{R_E^T(x(t))}{H_{1\times M} R_E(x(t))}$ is zero, or $\overline{R}_{x_k}$ does not exist, which is contradictory to the definition of FRM modeling. Hence, there never exists $x = 0$ to make $f_{FRM}(x) = 0$.

Theorem 5.1 can be extended to vector-valued functions by the following theorem:

Theorem 5.2 (**Universal approximation by MIMO FRMs**): For a continuous multi-variable function $G(x) = \left[g^1(x) \quad g^2(x) \quad \cdots \quad g^m(x)\right]^T \in \Re^m$ on $\tilde{E}$, $x = (x_1, x_2, \cdots, x_n) \in \tilde{E}$, given $\varepsilon > 0$, there exists a fuzzy system $f_{FRM}(x)$ in the form of Eq. (5.6), such that

$$\max_{k=1}^{m}(\sup_{x\in\tilde{E}}\left\| f_{FRM}^k(x) - g^k(x)\right\|) < \varepsilon . \tag{5.9}$$

Such a fuzzy system $[f_{FRM}^1(x), f_{FRM}^2(x), \cdots, f_{FRM}^m(x)]^T$ is called a universal approximator.

Proof: According to Theorem 5.1, there exists a fuzzy system $f_{FRM}^k(x)$, $k = 1, 2, \cdots, m$, in the form of Eq. (5.5), such that

$$\sup_{x\in\tilde{E}}\left\| f_{FRM}^k(x) - g^k(x)\right\| < \varepsilon ,\ k = 1, 2, \cdots, m . \tag{5.10}$$

Thus, $f_{FRM}(x) - G(x) = \begin{pmatrix} f_{FRM}^1(x) - g^1(x) \\ f_{FRM}^2(x) - g^2(x) \\ \vdots \\ f_{FRM}^m(x) - g^m(x) \end{pmatrix} = \begin{pmatrix} \overline{R}_{x_1}C_1 - g^1(x) \\ \overline{R}_{x_2}C_2 - g^2(x) \\ \vdots \\ \overline{R}_{x_m}C_m - g^m(x) \end{pmatrix}$. Eq. (5.9) can be proved.

5.3 Design of FRMs as Universal Approximators

According to Theorems 5.1 and 5.2, the design of FRM systems will be discussed in this section.

5.3.1 Design of FRM systems with two inputs and single output

Assume that $g(x)$ is a continuous function on a set $E = E_1 \times E_2 = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \in \Re^2$. The following steps are undertaken to formulate a fuzzy system to approximate $g(x)$.

Step 1: Given inputs x_i, define N_i fuzzy sets $X_i^1, X_i^2, \cdots, X_i^{N_i}$ in $[\alpha_i, \beta_i]$, which are normal, consistent, and complete with the corresponding MFs: $\mu_{X_i^1}(x_i; a_i^1, b_i^1, c_i^1, d_i^1)$, ..., $\mu_{X_i^{N_i}}(x_i; a_i^{N_i}, b_i^{N_i}, c_i^{N_i}, d_i^{N_i})$, where $i = 1, 2$, $X_i^1 < X_i^2 < \cdots < X_i^{N_i}$, $a_i^1 = b_i^1 = \alpha_i$ and $c_i^{N_i} = d_i^{N_i} = \beta_i$. Let $e_1^1 = \alpha_1$, $e_1^{N_1} = \beta_1$, and $e_1^j = \frac{1}{2}(b_1^j + c_1^j)$; $e_2^1 = \alpha_2$, $e_2^{N_2} = \beta_2$, and $e_2^j = \frac{1}{2}(b_2^j + c_2^j)$ for $j = 2, 3, \cdots, N_2 - 1$. Fig. 5.2 shows an example with $N_1 = 4$, $N_2 = 3$, $\alpha_1 = \alpha_2 = 0$, $\beta_1 = \beta_2 = 1$ with pseudo-trapezoid MFs.

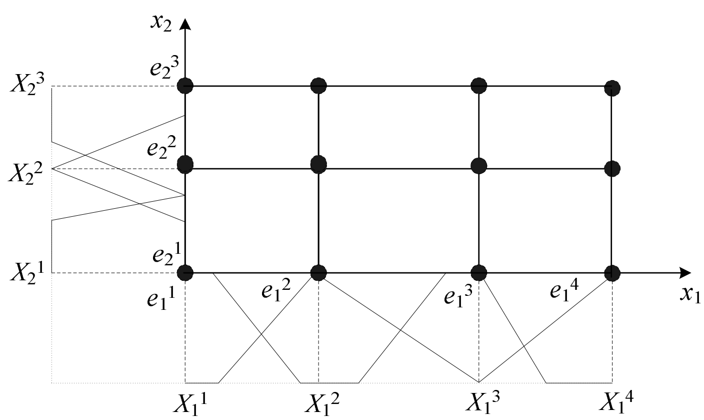


Fig. 5.2. An example with $N_1 = 4$, $N_2 = 3$, $\alpha_1 = \alpha_2 = 0$, $\beta_1 = \beta_2 = 1$.

Step 2: Construct $M = N_1 \times N_2$ fuzzy IF-THEN rules:

$$R^{i_1,i_2}: \; IF \; (x_1 \; is \; X_1^{i_1}) \; AND \; (x_2 \; is \; X_2^{i_2}), \; THEN \; (y \; is \; Y^{i_1,i_2}), \tag{5.11}$$

where $c^{i_1,i_2} = g(e_1^{i_1}, e_2^{i_2})$ is chosen as the center of the fuzzy sets Y^{i_1,i_2}, $i_1 = 1, 2, \cdots, N_1$, $i_2 = 1, 2, \cdots, N_2$.

Step 3: Formulate the FRM for the fuzzy system $f_{FRM}(x)$ of Eq. (5.11) using a max-product fuzzy operator and a center average defuzzifier:

$$f_{FRM}(x)=\frac{(V_{E_1}(x_1)\rhd V_{E_2}(x_2))^T(c^{11},c^{12},\cdots,c^{1N_2},\ \cdots\ ,c^{N_11},c^{N_12},\cdots,c^{N_1N_2})^T}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}(\mu_{X_1^{i_1}}(x_1)\mu_{X_2^{i_2}}(x_2))}=\frac{R_x^TC}{H_{1\times M_j}R_x}=\overline{R}_XC, \quad (5.12)$$

where $V_{E_1}(x_1)=(\mu_{X_1^1}(x_1),\cdots,\mu_{X_1^{N_1}}(x_1))^T$, $V_{E_2}(x_2)=(\mu_{X_2^1}(x_2),\cdots,\mu_{X_2^{N_2}}(x_2))^T$,

$V_E=V_{E_1}(x_1)\rhd V_{E_2}(x_2)$, $C=(c^{11},c^{12},\cdots,c^{1N_2},\ \cdots\ ,c^{N_11},c^{N_12},\cdots,c^{N_1N_2})^T\in\Re^{N_1N_2\times1}$.

Since the fuzzy sets $X_i^1,X_i^2,\cdots,X_i^{N_i}$ are complete, for each $x_i\in E_i$, $i=1,2,\cdots,n$, there exists at least one i_1 and one i_2, such that $\mu_{X_1^{i_1}}(x_1)\mu_{X_2^{i_2}}(x_2)\neq0$. Hence, the fuzzy system is well defined, or the denominator in Eq. (5.12) is not zero.

Step 4: Compute the output of the fuzzy system $f_{FRM}(x)$ and compare it with the values of $g(x)$ at $x=(e_1^{i_1},e_2^{i_2})$ for $i_1=1,2,\cdots,N_1$, $i_2=1,2,\cdots,N_2$, which is the resulting FRM system with two inputs and one output $f_{FRM}(x)$ at $x=(e_1^{i_1},e_2^{i_2})$.

5.3.2 Design of MISO FRM systems

Assume that $g(x)$ is a continuous function on a compact set $\widetilde{E}=[\alpha_1,\beta_1]\times[\alpha_2,\beta_2]\times\cdots\times[\alpha_n,\beta_n]$. Given inputs $x_i\in[\alpha_i,\beta_i]\subset\Re$, $i=1,2,\cdots,n$, define N_i fuzzy sets $X_i^1,X_i^2,\cdots,X_i^{N_i}$ in $[\alpha_i,\beta_i]$, then construct $M=N_1\times N_2\times\cdots\times N_n$ fuzzy IF-THEN rules in the following form:

$$R^l:\ IF\ (x_1\ is\ X_1^{p_1})\ AND\ \cdots AND\ (x_n\ is\ X_n^{p_n}),\ THEN\ (y\ is\ Y^l),\ l=1,2,\cdots,M, \quad (5.13)$$

where M is the number of fuzzy rules; x_i is the i-th input variable on the universe of discourse E_i; $X_i^1,\cdots,X_i^{N_i}$ are fuzzy sets defined on E_i; N_i is the number of fuzzy sets, $i=1,2,\cdots,n$, $p_i\in\{1,2,\cdots,N_i\}$, $l=p_1+\sum_{i=2}^{n}[(p_i-1)\prod_{j=1}^{i-1}N_j]$; y is the output variable on the universe of discourse F; $Y^1,\cdots,Y^M$ are fuzzy sets defined on F. $c^l=g(e_1^{p_1},e_2^{p_2},\cdots,e_n^{p_n})$ is chosen as the centers of the fuzzy sets Y^l.

By using similar steps as in Section 4.3.2 with two inputs one output FRM systems, the following FRM for the fuzzy system $f_{FRM}(x)$ can be formulated of Eq. (5.13):

$$f_{FRM}(x) = \frac{R_E^T(x, y)C}{H_{1\times M} R_E(x, y)} = \overline{R}_x C, \tag{5.14}$$

where $\overline{R}_x = \frac{R_E^T(x, y)}{H_{1\times M} R_E(x, y)}$, $C = (c^1, \ \cdots, \ c^{N_1 \cdots N_n})^T \in \Re^{M\times 1}$, $H_{1\times M} = (1 \quad 1 \quad \cdots \quad 1) \in \Re^{1\times M}$,

$R_E(x, y) = (\mu_{Y^1}(y), \cdots, \mu_{Y^M}(y))^T = M_R \triangleright V_E(x)$, $V_E = V_{E_1}(x_1) \triangleright V_{E_2}(x_2) \triangleright \cdots \triangleright V_{E_n}(x_n)$,

$V_{E_1}(x_1) = (\mu_{X_1^1}(x_1), \cdots, \mu_{X_1^{N_1}}(x_1))^T$, ... , $V_{E_n}(x_n) = (\mu_{X_n^1}(x_n), \cdots, \mu_{X_n^{N_n}}(x_n))^T$.

M_R is the FRM of the fuzzy system in Eq. (5.13). There exists at least a group of $\{ i_1, i_2, \cdots, i_n \}$ such that $\mu_{X_1^{P1}}(x_1)\mu_{X_2^{P2}}(x_2)\cdots\mu_{X_n^{Pn}}(x_n) \neq 0$. Hence, the fuzzy system is well defined.

5.3.3 Design of MIMO FRM systems

Consider a multi-variable function $G(x) = [g^1(x) \ g^2(x) \ \cdots \ g^m(x)]^T$ with continuous $g^k(x)$ on $\tilde{E}$. The MISO FRM model $f^k{}_{FRM}(x) \in F_{FRM}(\tilde{E})$ can be formulated using Eq. (5.14). The fuzzy universal approximator can then be obtained as

$$[f_{FRM}^1(x), f_{FRM}^2(x), \cdots, f_{FRM}^m(x)]^T.$$

5.4 Approximation Accuracy of the FRM

The approximation accuracy of the $f_{FRM}(x)$ will be investigated in this section.

5.4.1 Approximation accuracy of an FRM with two inputs single output FRM

Theorem 5.3: If $f_{FRM}(x)$ is the fuzzy system as in Eq. (5.12), and $g(x)$ is an unknown but differentiable function on $E = E_1 \times E_2 = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \subset \Re^2$, then

$$\| f_{FRM}(x) - g(x) \|_\infty \leq \left\| \frac{\partial g}{\partial x_1} \right\|_\infty h_1 + \left\| \frac{\partial g}{\partial x_2} \right\|_\infty h_2, \tag{5.15}$$

where the infinite norm $\|*\|_\infty$ stands for $\|d(x)\|_\infty = \sup_{x\in E}|d(x)|$, and $h_i = \max_{1\le j\le N_i-1}\left|e_i^{j+1} - e_i^j\right|$, $i = 1, 2$.

Proof: Let $E^{i_1 i_2} = [e_1^{i_1}, e_1^{i_1+1}]\times[e_2^{i_2}, e_2^{i_2+1}]$, $i_1 = 1, 2, \cdots, N_1 - 1$, and $i_2 = 1, 2, \cdots, N_2 - 1$. Since $[\alpha_i, \beta_i] = [e_i^1, e_i^2]\bigcup[e_i^2, e_i^3]\bigcup\cdots\bigcup[e_i^{N_i-1}, e_i^{N_i}]$, $i = 1, 2$, then

$$E = E_1\times E_2 = [\alpha_1, \beta_1]\times[\alpha_2, \beta_2] = \bigcup_{i_1=1}^{(N_1-1)}\bigcup_{i_2=1}^{(N_2-1)} E^{i_1 i_2}.$$

Therefore, given any $x_1 \in E_1$, $x_2 \in E_2$, there exists one $E^{i_1 i_2} = E^{i_1}\times E^{i_2}$ such that $(x_1, x_2)\in E^{i_1 i_2}$, $x_1 \in [e_1^{i_1}, e_1^{i_1+1}]$, and $x_2 \in [e_2^{i_2}, e_2^{i_2+1}]$.

Because the fuzzy sets $X_1^1, X_1^2, \cdots, X_1^{N_1}$ are normal, consistent and complete, at least one and at most two $\mu_{X_1^{j_1}}(x_1)$ are nonzero for $j_1 = 1, 2, \cdots, N_1$. According to the definition of $e_1^{j_1}$, the two possible nonzero MFs for $\mu_{X_1^{j_1}}(x_1)$, $j_1 = 1, 2, \cdots, N_1$, are $\mu_{X_1^{i_1}}(x_1)$ and $\mu_{X_1^{i_1+1}}(x_1)$. Similarly, the two possible nonzero MFs for $\mu_{X_2^{j_2}}(x_2)$, $j_2 = 1, 2, \cdots, N_2$, are $\mu_{X_2^{i_2}}(x_2)$ and $\mu_{X_2^{i_2+1}}(x_2)$. Hence, the fuzzy system $f_{FRM}(x)$ of Eq. (5.11) can be simplified as Eq. (5.16).

$$|f_{FRM}(x) - g(x)| =$$

$$\frac{[(0\cdots0\,\mu_{X_1^{i_1}}(e_1^{i_1}), \mu_{X_1^{i_1+1}}(e_1^{i_1+1})\,0\cdots0)^T \triangleright (0\cdots0\,\mu_{X_2^{i_2}}(e_2^{i_2}), \mu_{X_2^{i_2+1}}(e_2^{i_2+1})\,0\cdots0)^T]^T(c^{11}, c^{12}, \cdots, c^{1N_2}, \cdots, c^{N_1 1}, c^{N_1 2}, \cdots, c^{N_1 N_2})^T}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}(\mu_{X_1^{i_1}}(x_1)\mu_{X_2^{i_2}}(x_2))}$$

$$= \frac{(0\cdots0\min(\mu_{X_1^{i_1}}(e_1^{i_1}), \mu_{X_2^{i_2}}(e_2^{i_2})), \min(\mu_{X_1^{i_1}}(e_1^{i_1}), \mu_{X_2^{i_2+1}}(e_2^{i_2+1}))\,0\cdots0\min(\mu_{X_1^{i_1+1}}(e_1^{i_1+1}), \mu_{X_2^{i_2}}(e_2^{i_2}))\min(\mu_{X_1^{i_1+1}}(e_1^{i_1+1}), \mu_{X_2^{i_2+1}}(e_2^{i_2+1}))\,0\cdots0)\cdot(c^{11}, c^{12}, \cdots, c^{1N_2}, \cdots, c^{N_1 1}, c^{N_1 2}, \cdots, c^{N_1 N_2})^T}{\sum_{j_1=i_1}^{i_1+1}\sum_{j_2=i_2}^{i_2+1}[\min(\mu_{X_1^{j_1}}(x_1), \mu_{X_2^{j_2}}(x_2))]}$$

$$= \frac{\min(\mu_{X_1^{i_1}}(e_1^{i_1}), \mu_{X_2^{i_2}}(e_2^{i_2}))c^{i_1 i_2} + \min(\mu_{X_1^{i_1}}(e_1^{i_1}), \mu_{X_2^{i_2+1}}(e_2^{i_2+1}))c^{i_1(i_2+1)} + \min(\mu_{X_1^{i_1+1}}(e_1^{i_1+1}), \mu_{X_2^{i_2}}(e_2^{i_2}))c^{(i_1+1)i_2} + \min(\mu_{X_1^{i_1+1}}(e_1^{i_1+1}), \mu_{X_2^{i_2+1}}(e_2^{i_2+1})c^{(i_1+1)(i_2+1)}}{\sum_{j_1=i_1}^{i_1+1}\sum_{j_2=i_2}^{i_2+1}[\min(\mu_{X_1^{j_1}}(x_1), \mu_{X_2^{j_2}}(x_2))]}$$

$$= \frac{\min(\mu_{X_1^{i_1}}(e_1^{i_1}), \mu_{X_2^{i_2}}(e_2^{i_2}))g(e_1^{i_1}, e_2^{i_2}) + \min(\mu_{X_1^{i_1}}(e_1^{i_1}), \mu_{X_2^{i_2+1}}(e_2^{i_2+1}))g(e_1^{i_1}, e_2^{i_2+1}) + \min(\mu_{X_1^{i_1+1}}(e_1^{i_1+1}), \mu_{X_2^{i_2}}(e_2^{i_2}))g(e_1^{i_1+1}, e_2^{i_2}) + \min(\mu_{X_1^{i_1+1}}(e_1^{i_1+1}), \mu_{X_2^{i_2+1}}(e_2^{i_2+1})g(e_1^{i_1+1}, e_2^{i_2+1})}{\sum_{j_1=i_1}^{i_1+1}\sum_{j_2=i_2}^{i_2+1}\mu_{X_1^{j_1}}(x_1)\cdot\mu_{X_2^{j_2}}(x_2)}$$

$$\le \sum_{j_1=i_1}^{i_1+1}\sum_{j_2=i_2}^{i_2+1}\left[\frac{\mu_{X_1^{j_1}}(x_1)\cdot\mu_{X_2^{j_2}}(x_2)}{\sum_{j_1=i_1}^{i_1+1}\sum_{j_2=i_2}^{i_2+1}\mu_{X_1^{j_1}}(x_1)\mu_{X_2^{j_2}}(x_2)}\right]\cdot\left|g(x) - g(e_1^{j_1}, e_2^{j_2})\right|$$

$$\le \max_{j_1=i_1, i_1+1;\, j_2=i_2, i_2+1}\left|g(x) - g(e_1^{j_1}, e_2^{j_2})\right|. \qquad (5.16)$$

Using the mean value theorem [32], Eq. (5.16) becomes:

$$\left|f_{FRM}(x)-g(x)\right| \le \max_{j_1=i_1,i_1+1;\,j_2=i_2,i_2+1}\left(\left\|\frac{\partial g}{\partial x_1}\right\|_\infty \left|x_1-e_1^{j_1}\right| + \left\|\frac{\partial g}{\partial x_2}\right\|_\infty \left|x_2-e_2^{j_2}\right|\right). \tag{5.17}$$

Since $(x_1, x_2) \in E^{i_1 i_2}$, it follows that $x_1 \in [e_1^{i_1}, e_1^{i_1+1}]$ and $x_2 \in [e_2^{i_2}, e_2^{i_2+1}]$. Thus,

$$\left|x_1-e_1^{j_1}\right| \le \left|e_1^{i_1+1}-e_1^{i_1}\right| \text{ and } \left|x_2-e_2^{j_2}\right| \le \left|e_2^{i_2}-e_2^{i_2+1}\right|,$$

for $j_1 = i_1,\ i_1+1$ and $j_2 = i_2,\ i_2+1$. Thus, Eq. (5.17) becomes

$$\left|g(x)-f_{FRM}(x)\right| \le \left\|\frac{\partial g}{\partial x_1}\right\|_\infty \left|e_1^{i_1+1}-e_1^{i_1}\right| + \left\|\frac{\partial g}{\partial x_2}\right\|_\infty \left|e_2^{i_2+1}-e_2^{i_2}\right|, \text{ or}$$

$$\left|g(x)-f_{FRM}(x)\right| \le \left\|g(x)-f_{FRM}(x)\right\|_\infty = \sup_{X\in E}\left|g(x)-f_{FRM}(x)\right|$$

$$\le \left\|\frac{\partial g}{\partial x_1}\right\|_\infty \max_{1\le i_1\le N_1-1}\left|e_1^{i_1+1}-e_1^{i_1}\right| + \left\|\frac{\partial g}{\partial x_2}\right\|_\infty \max_{1\le i_2\le N_2-1}\left|e_2^{i_2+1}-e_2^{i_2}\right|$$

$$\le \left\|\frac{\partial g}{\partial x_1}\right\|_\infty h_1 + \left\|\frac{\partial g}{\partial x_2}\right\|_\infty h_2 .$$

According to Eq. (5.15) in Theorem 5.3, it can be concluded that the fuzzy system Eq. (5.11) is a universal approximator.

5.4.2 Approximation accuracy of MISO FRM systems

Theorem 5.4: If $f_{FRM}(x)$ is a fuzzy system as in Eq. (5.14) and $g(x)$ is unknown but differentiable on $\tilde{E}$, then

$$\left\|f_{FRM}(x)-g(x)\right\|_\infty \le \left\|\frac{\partial g}{\partial x_1}\right\|_\infty h_1 + \left\|\frac{\partial g}{\partial x_2}\right\|_\infty h_2 + \cdots + \left\|\frac{\partial g}{\partial x_n}\right\|_\infty h_n , \tag{5.18}$$

where the infinite norm $\|*\|_\infty$ stands for $\|d(x)\|_\infty = \sup_{x\in\tilde{E}}|d(x)|$, and $h_i = \max_{1\le j\le N_i-1}\left|e_i^{j+1}-e_i^j\right|,\ i=1,2,\cdots,n$.

Its proof is similar to that of Theorem 5.3.

5.4.3 Approximation accuracy of MIMO FRM systems

Theorem 5.5: If $f_{FRM}^{k}(x),\quad k=1,2,\cdots,m$, are the fuzzy systems as in Eq. (5.14), and an unknown function $G(x)=\left[g^{1}(x)\ \ g^{2}(x)\ \ \cdots\ \ g^{m}(x)\right]^{T}$ is differentiable on $\tilde{E}$, then

$$\left\|\begin{matrix} f_{FRM}^{1}(x)-g^{1}(x) \\ f_{FRM}^{2}(x)-g^{2}(x) \\ \vdots \\ f_{FRM}^{m}(x)-g^{m}(x) \end{matrix}\right\|_{\infty} \le \left\|\frac{\partial G(x)}{\partial x_1}\right\|_{\infty} h_1 + \left\|\frac{\partial G(x)}{\partial x_2}\right\|_{\infty} h_2 + \cdots + \left\|\frac{\partial G(x)}{\partial x_n}\right\|_{\infty} h_n, \tag{5.19}$$

where the infinite norm $\|*\|_{\infty}$ is defined as $\|d(x)\|_{\infty}=\sup_{x_i\in E_i}|d(x_i)|$, and $h_i=\max_{1\le j\le N_i-1}\left|e_i^{j+1}-e_i^{j}\right|,\ i=1,2,\cdots,n$.

Proof: As

$$\left\|\begin{matrix} f_{FRM}^{1}(x)-g^{1}(x) \\ f_{FRM}^{2}(x)-g^{2}(x) \\ \vdots \\ f_{FRM}^{m}(x)-g^{m}(x) \end{matrix}\right\|_{\infty} = \max_{k=1}^{m}\left\|f_{FRM}{}^{k}(x)-g^{k}(x)\right\|_{\infty},$$

from Theorem 4.4,

$$\left\|f_{FRM}^{k}(x)-g^{k}(x)\right\|_{\infty} \le \left\|\frac{\partial g^{k}}{\partial x_1}\right\|_{\infty} h_1 + \left\|\frac{\partial g^{k}}{\partial x_2}\right\|_{\infty} h_2 + \cdots + \left\|\frac{\partial g^{k}}{\partial x_n}\right\|_{\infty} h_n,\ k=1,2,\cdots,m.$$

Hence, $$\left\|\begin{matrix} f_{FRM}^{1}(x)-g^{1}(x) \\ f_{FRM}^{2}(x)-g^{2}(x) \\ \vdots \\ f_{FRM}^{m}(x)-g^{m}(x) \end{matrix}\right\|_{\infty} \le \max_{k=1}^{m}\left(\left\|\frac{\partial g^{k}}{\partial x_1}\right\|_{\infty} h_1 + \left\|\frac{\partial g^{k}}{\partial x_2}\right\|_{\infty} h_2 + \cdots + \left\|\frac{\partial g^{k}}{\partial x_n}\right\|_{\infty} h_n\right).$$

Because $\max_{k=1}^{m}\left(\left\|\frac{\partial g^{k}}{\partial x_1}\right\|_{\infty} h_1 + \left\|\frac{\partial g^{k}}{\partial x_2}\right\|_{\infty} h_2 + \cdots + \left\|\frac{\partial g^{k}}{\partial x_n}\right\|_{\infty} h_n\right)$

$$\le \left(\max_{k=1}^{m}\left\|\frac{\partial g^{k}}{\partial x_1}\right\|_{\infty}\right)h_1 + \left(\max_{k=1}^{m}\left\|\frac{\partial g^{k}}{\partial x_2}\right\|_{\infty}\right)h_2 + \cdots + \left(\max_{k=1}^{m}\left\|\frac{\partial g^{k}}{\partial x_n}\right\|_{\infty}\right)h_n,$$

the following inequality is true:

$$\left\| \begin{matrix} f_{FRM}^{1}(x) - g^{1}(x) \\ f_{FRM}^{2}(x) - g^{2}(x) \\ \vdots \\ f_{FRM}^{m}(x) - g^{m}(x) \end{matrix} \right\|_{\infty} \le \left(\max_{k=1}^{m} \left\| \frac{\partial g^{k}}{\partial x_{1}} \right\|_{\infty} \right) \cdot h_{1} + \left(\max_{k=1}^{m} \left\| \frac{\partial g^{k}}{\partial x_{2}} \right\|_{\infty} \right) \cdot h_{2} + \cdots + \left(\max_{k=1}^{m} \left\| \frac{\partial g^{k}}{\partial x_{n}} \right\|_{\infty} \right) \cdot h_{n} \cdot$$

Since $\max_{k=1}^{m} \left\| \frac{\partial g^{k}}{\partial x_{i}} \right\|_{\infty} = \left\| \frac{\partial G(x)}{\partial x_{i}} \right\|_{\infty}$, $i = 1, 2, \cdots, n$, it can be verified that

$$\left\| \begin{matrix} f_{FRM}^{1}(x) - g^{1}(x) \\ f_{FRM}^{2}(x) - g^{2}(x) \\ \vdots \\ f_{FRM}^{m}(x) - g^{m}(x) \end{matrix} \right\|_{\infty} \le \left\| \frac{\partial G(x)}{\partial x_{1}} \right\|_{\infty} \cdot h_{1} + \left\| \frac{\partial G(x)}{\partial x_{2}} \right\|_{\infty} \cdot h_{2} + \cdots + \left\| \frac{\partial G(x)}{\partial x_{n}} \right\|_{\infty} \cdot h_{n} \cdot$$

5.5 Numerical Simulations

The effectiveness of the proposed FRM design technique will be examined by simulation in this section.

Example 5.1:

Suppose a continuous function $g(x) = \sin(x)$ is defined on $E = [-3, 3]$; a fuzzy system with FRM model will be designed to uniformly approximate $g(x)$, with a required accuracy $\varepsilon = 0.2$, that is

$$\sup_{x \in E} \left| \sin(x) - f_{FRM}(x) \right| < \varepsilon .$$

Then, since $\left\| \frac{\partial g}{\partial x} \right\|_{\infty} = \left\| \cos(x) \right\|_{\infty} = \sup_{x \in [-3, 3]} \left| \cos x \right| = 1$, it follows from Eq. (5.15) that the fuzzy system with $h = 0.2$ can satisfy the related requirement in Theorem 5.3. Fig. 5.3 illustrates an example with triangular MFs X^{j} for the input with 31 fuzzy sets, $j = 1, 2, \cdots, 31$, which are defined as

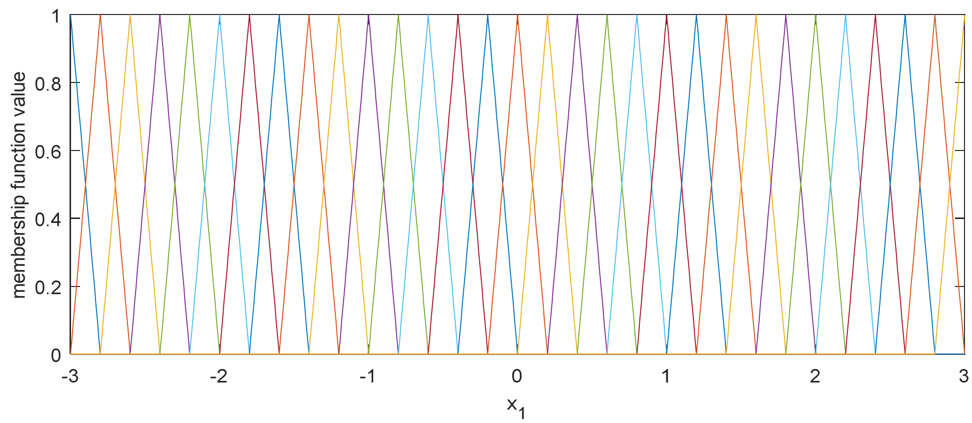


Fig. 5.3. The membership functions of fuzzy sets for an input variable.

$\mu_{X^1}(x) = \mu_{X^1}(x;\ -3, -3, -2.8)$,

$\mu_{X^{31}}(x) = \mu_{X^{31}}(x;\ 2.8,\ 3,\ 3)$,

$\mu_{X^j}(x) = \mu_{X^j}(x; e^{j-1}, e^j, e^{j+1})$,

where $j = 2, 3, \cdots, 30$, $e^j = -3 + 0.2(j-1)$.

The center of B^j can be selected as $c^j = g(e^j) = \sin(e^j)$.

The fuzzy rules are defined as

$$R^j : IF\ \ x_1\ \ is\ \ X^j,\ THEN\ \ y\ \ is\ \ Y^j,\ j = 1, 2, \cdots, 31, \tag{5.20}$$

where Y^j is the fuzzy set of the output y.

Then, $V_E(x) = V_{E_1}(x_1) = (\mu_{X^1}(x), \mu_{X^2}(x), \cdots, \mu_{X^{31}}(x))^T$,

$R_E(x) = V_{F_1}(y_1) = (\mu_{Y^1}(y), \mu_{Y^2}(y), \cdots, \mu_{Y^{31}}(y))^T == M_R \triangleright V_E(x)$,

where M_R is the FRM of the fuzzy system in Eq. (5.20), $H_{1\times M} = (1,\ 1, \cdots,\ \ 1) \in M_{1\times 31}$, and $C = (c_1, c_2, \cdots, c_{31})^T$.

According to Eq. (5.11), the designed fuzzy system is $f_{FRM}(x) = \overline{R}_x C$, where $\overline{R}_x = \dfrac{R_E^T(x)}{H_{1\times M} R_E(x)}$. The exact estimated output of $f_{FRM}(x) = \overline{R}_x C$ is plotted in Fig. 5.4 (a). It is seen from Fig. 5.4 (a) that $f_{FRM}(x)$ and $g(x)$ are almost identical, with a mean absolute error of 0.002 (from Fig. 5.4(b)) between the real output and the FRM model output.

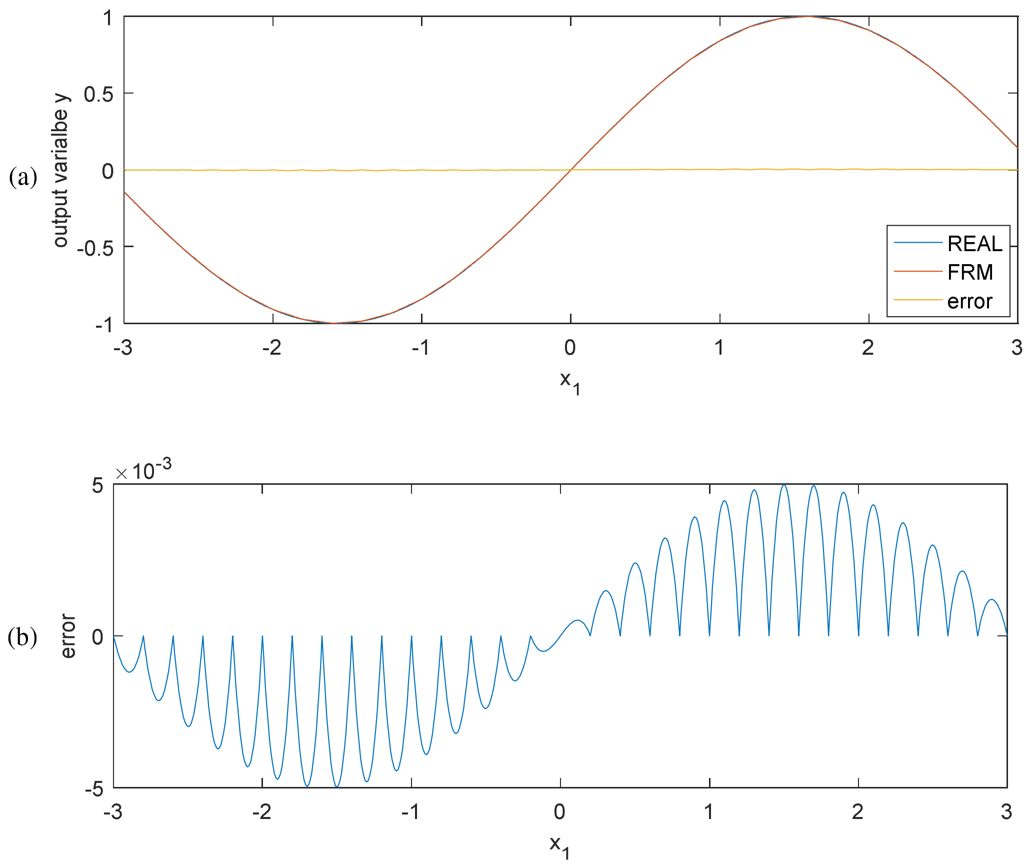


Fig. 5.4. (a) The real output and FRM model output. (b) The error between these two outputs.

Example 5.2:

Suppose the continuous function is $g(x_1, x_2) = 0.52 + 0.1x_1 + 0.28x_2 - 0.06x_1x_2$

with a required accuracy $\varepsilon = 0.1$, or $\sup_{x\in E}\|g(x_1, x_2) - f_{FRM}(x_1, x_2)\| < \varepsilon$.

Then, since $\left\|\frac{\partial g}{\partial x_1}\right\|_\infty = \|0.1 - 0.06x_2\|_\infty = \sup_{x_2\in[-1,1]}|0.1 - 0.06x_2| = 0.16$, and $\left\|\frac{\partial g}{\partial x_2}\right\|_\infty =$ $= \|0.28 - 0.06x_1\|_\infty = \sup_{x_1\in[-1,1]}|0.28 - 0.06x_1| = 0.34$, it follows from Eq. (5.12) that

$h_1 = h_2 = 0.2$, and

$$\left\| g(x_1, x_2) - f_{FRM}(x_1, x_2) \right\|_\infty \le 0.16 \times 0.2 + 0.34 \times 0.2 = 0.1 .$$

If X_i^j has 11 fuzzy sets defined in $[-1,\ 1]$, $j = 1, 2, \cdots, 11$, then the following $11 \times 11 = 121$ fuzzy rules can be defined:

$$R^l : IF\ \ (x_1\ \ is\ \ X_1^{i_1}) \text{ and } (x_2\ \ is\ \ X_2^{i_2}),\ \ THEN\ \ (y\ \ is\ \ Y^{i_1 i_2}), \tag{5.21}$$

where $X_1^{i_1}$, $X_2^{i_2}$ and $Y^{i_1 i_2}$ are the fuzzy sets of the respective inputs and outputs, $l = 1, 2, \cdots, 121$, $i_1, i_2 = 1, 2, \cdots, 11$.

If $c^l = g(e^{i_1}, e^{i_2}) = 0.52 + 0.1e^{i_1} + 0.28e^{i_2} - 0.06e^{i_1}e^{i_2}$ is the center of Y^l, then the triangular MFs X_i^j can be determined as:

$$\mu_{X_i^1}(x_i) = \mu_{X_i^1}(x_i; -1, -1, -0.8),$$

$$\mu_{X_i^{11}}(x_i) = \mu_{X_i^{11}}(x_i; 0.8, 1, 1),$$

$$\mu_{X^j}(x_i) = \mu_{X^j}(x_i;\ e^{j-1},\ e^j,\ e^{j+1}),$$

where $e^j = -1 + 0.2(j-1)$, $j = 2, 3, \cdots, 10$.

Then, $V_{E_i}(x_i) = (\mu_{X_i^1}(x), \mu_{X_i^2}(x), \cdots, \mu_{X_i^{11}}(x))^T$, $i = 1,\ \ 2$, $V_E(x) = V_{E_1}(x_1) \triangleright V_{E_2}(x_2)$,

$V_F(y) = V_{F_1}(y_1) = (\mu_{Y^1}(y), \mu_{Y^2}(y), \cdots, \mu_{Y^{31}}(y))^T = M_R \triangleright V_E(x)$,

where M_R is the FRM of the fuzzy system in Eq. (5.21).

In using some sampling data in simulation, by Eq. (5.5) the designed fuzzy system becomes

$$f_{FRM}(x_1, x_2) = \overline{R}_x C .$$

where $C = (c_1, c_2, \cdots, c_{121})^T$ and $\overline{R}_x = \dfrac{R_E^T(x)}{H_{1\times M} R_E(x)}$ with $H_{1\times M} = (1,\ 1, \cdots,\ \ 1) \in M_{1\times 121}$.

It is seen from Fig. 5.5 that $f_{FRM}(x_1, x_2)$ and $g(x_1, x_2)$ are almost identical and the image of errors between the real outputs and FRM model outputs that are almost zero (3.174×10^{-7}).

From Examples 5.1 and 5.2, the effectiveness of universal approximation can be verified.

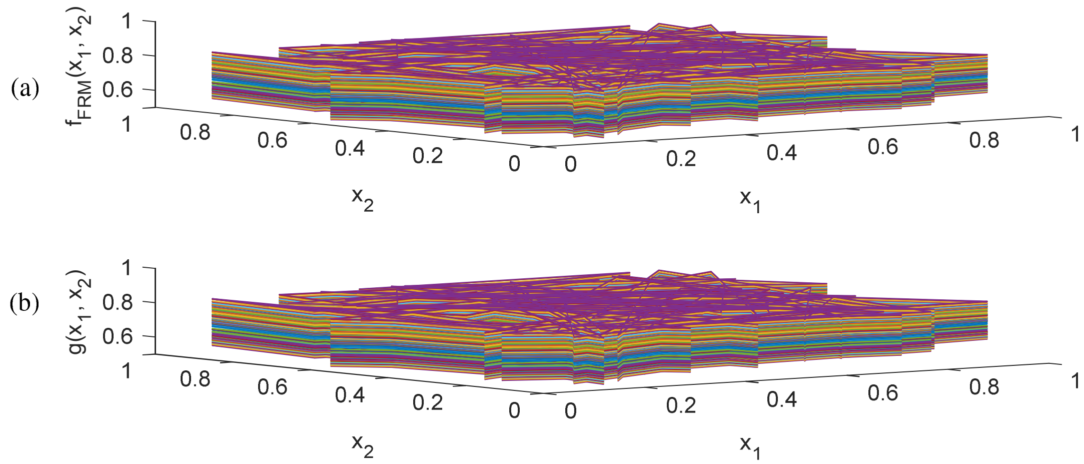


Fig. 5.5. (a) FRM model outputs; (b) The real outputs.

5.6 Concluding Remarks

Based on FRM modeling, a universal approximator has been proposed in this chapter for MIMO fuzzy system modeling. The STP of logic matrices is employed to expand fuzzy reasoning operations and the fuzzy input and output variables are represented vectors. The analytical proof is provided to theoretically evaluate the design algorithm and approximation of FRM systems as a universal approximator. The effectiveness of the proposed FRM approximation technique is verified by some simulation examples. The related contribution has been published in

(1) H. Lyu, W. Wang, X. Liu, "Universal approximation of multi-variable fuzzy systems by semi-tensor product", *IEEE Transactions on Fuzzy Systems*, Vol. 28, No.11, pp. 2972-2981, Nov. 2020.

Advanced research will be undertaken in Chapters 6 and 7 to develop a framework to facilitate the implementation of the proposed MIMO universal approximation theory.

Chapter 6 Parameters Training of FRM Models

The modeling accuracy depends on the parameters of FRM models, such as the number of fuzzy variables, fuzzy sets and fuzzy rules, the center points and shapes of membership functions (MFs). As discussed in Chapter 4, two modeling methods have been proposed to develop primary FRM models based on the sampling database. In the direct modeling the parameters are assumed to be tuned manually by trial and error. When the centers in output fuzzy sets are unknown, they can be adjusted by the LSE algorithm in indirect FRM modeling. However, other parameters cannot be optimized in both modeling methods for a fuzzy system based on FRM models. Hence, the objective of this chapter is to train other parameters in FRM models to improve modeling accuracy of MIMO fuzzy systems. A new neural-fuzzy (NF) network based on the FRM and fuzzy STP is proposed, which are then trained using the LSE and the recursive Levenberg-Marquaedt (RLM) algorithms.

The remainder of this chapter is organized as follows. The preliminaries about MIMO FRM Systems are introduced in Section 6.1. Section 6.2 discusses the structure identification of FRMs by using the NF network. The FRM model parameter identification and training are discussed in Section 6.3. The effectiveness of the proposed FRM identification technique is tested by simulations in Section 6.4.

6.1 Preliminary

According to the definitions of FRMs in Chapters 3 and the STP of logic matrices proposed in Chapters 2, a general multi-variable FRM model can be constructed by two modeling methods using the sampling training data pairs as discussed in Chapter 4. If a training database can be obtained for the system of interest, the fuzzy relation function $M_R(x, y) := f_{FRM}(x_1, \cdots, x_n; y_1, \cdots, y_m)$ will be established based on the FRM system identification instead of the linguistic reasoning.

Consider a general n-inputs and m-outputs FRM model with the following fuzzy rules:

$$R^l: IF \quad (x_1 \quad is \quad X_1^{p_1}) \quad AND \cdots AND \quad (x_n \quad is \quad X_n^{p_n}),$$

$$THEN \quad (y_1 \quad is \quad Y_1^l) \quad AND \cdots AND \quad (y_m \quad is \quad Y_m^l), \tag{6.1}$$

where all the parameters are assumed to be the same as in Eq. (5.1), including the centers c_k^l, $l = 1, \cdots, M$, $k = 1, 2, \cdots, m$.

Given an input $x_0 = (x_{10}, x_{20}, \cdots, x_{n0})$, $C_k = (c_k^1, \cdots, c_k^M)^T$, and an initial FRM model $M_R(x, y)$ as constructed using the method in Section 4.3, the estimated output y_{k0} can be computed as

$$y_{k0} := f_{FRM}^k (x_{10}, x_{20}, \cdots, x_{n0}) = \overline{R}_{x_{k0}} C_k, \tag{6.2}$$

which can be rewritten as

$$y_{k0} = f_{FRM}^k (x) = \overline{R}_{x_{k0}} C_k, \tag{6.3}$$

where $x_0 = (x_{10}, x_{20}, \cdots, x_{n0})$, $\overline{R}_{x_{k0}} = \dfrac{R_E^T(x_0, y_{k0})}{H_{1\times M} R_E(x_0, y_{k0})}$, $H_{1\times M} = (1 \quad 1 \quad \cdots \quad 1) \in \Re^{1\times M}$, $C_k = (c_k^1, \cdots, c_k^M)^T$, $k = 1, 2, \cdots, m$.

However, these previous modeling techniques cannot optimize all parameters (especially nonlinear parameters) of the FRM model in Eq. (6.1). In next section, the parameters of both input and output variables will be optimized by a new NF network and a hybrid training method.

6.2 Architecture Modeling of Neural-Fuzzy Systems based on FRMs

To optimize system parameters in the FRM system, a new NF and STP network approach, NF-STP in short, will be proposed to identify the FRM model structure and train system parameters. Fig. 6.1 illustrates the equivalent network architecture. It has five layers to implement the related FRM and STP operations, as discussed below.

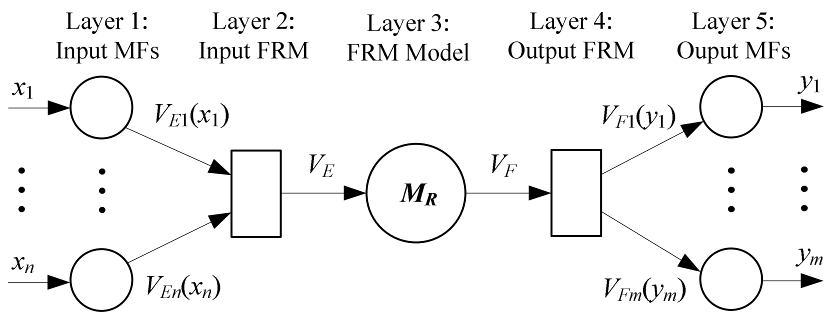


Fig. 6.1. The structure of a proposed NF-STP model.

Layer 1: This input layer consists of fuzzy input vectors $\{x_1, x_2, \cdots, x_n\}$, which are fuzzified by the corresponding input vector nodes with MFs $E_{x_i} = \{X_i^1, \cdots, X_i^{N_i}\}$, $i = 1, 2, \cdots, n$. The MFs can be in the forms of triangle, trapezoid, sigmoid, Gaussian, etc. The output of each node in this layer is a fuzzy vector expression $V_{E_i}(x_i) = (\mu_{X_i^1}(x_i), \cdots, \mu_{X_i^{N_i}}(x_i))^T$, $i = 1, 2, \cdots, n$.

Layer 2: The inputs to this layer are fuzzy variable vectors $V_{E_i}(x_i)$, $i = 1, 2, \cdots, n$ and the corresponding STP of fuzzy logic matrices. The output of this layer is the vector V_E to be inputted to the FRM, which can be calculated by the fuzzy STP operation $V_E := V_E\,(x_1, \cdots, x_n) = V_{E_1}(x_1) \triangleright V_{E_2}(x_2) \triangleright \cdots \triangleright V_{E_n}(x_n)$.

Layer 3: From the input V_E, the output of this layer is V_F, or the output of the FRM, $V_F := V_F(y_1, \cdots, y_m) = V_{F_1}(y_1) \triangleright V_{F_2}(y_2) \triangleright \cdots \triangleright V_{F_m}(y_m)$. The initial FRM model of the fuzzy system is formulated by the direct modeling method as discussed in Chapter 4. The training of the FRM model parameters $M_R(x, y)$ will be discussed in Section 6.3.

Layer 4: With the inputs V_F, the node output of this layer is the fuzzy variable vector $y_k \in \{y_1, y_2, \cdots, y_m\}$. When the FRM model $M_R(x, y)$ is identified, the output FRM $V_F(y_1, y_2, \cdots, y_m)$ can be calculated from the fuzzy STP operation and $V_F(y_1, y_2, \cdots, y_m) = V_{F_1}(y_1) \triangleright V_{F_2}(y_2) \triangleright \cdots \triangleright V_{F_m}(y_m)$.

Layer 5: Each output value is computed by the STP as the fuzzy reasoning operations in this layer, and center-average defuzzification based on the vector $V_{F_k}(y_k) = (\mu_{Y_k^1}(y_k), \cdots, \mu_{Y_k^M}(y_k))^T$, $k = 1,\ 2, \cdots, m$.

From the NF-STP identification operation as discussed above, the formulated FRM model is functionally equivalent to an FRM model, except that the NF-STP network can be used to train parameters for the FRM models. On the other hand, the structure here of the NF-STP model is not unique. For example, layers 2 and 3 can be combined into a more complex, but functionally equivalent layer.

6.3 Parameters Optimization based on the NF-STP Model

In general, the design of a fuzzy model includes processes of structure identification and parameter optimization. In this section, in the proposed NF-STP model in Section 6.2, a hybrid training method will be adopted to train the RFM model parameters. Without loss of generality, consider the following MIMO TSK FRM model [154]:

$$
\begin{aligned}
R^l:\ & IF\ \ (x_1\ \ is\ \ X_1^{p_1})\ \ AND \cdots AND\ \ (x_n\ \ is\ \ X_n^{p_n}), \\
& THEN\ \ (y_1 = c_{10}^l + c_{11}^l x_1 + c_{12}^l x_2 + \cdots c_{1n}^l x_n)\ AND\ (y_2 = c_{20}^l + c_{21}^l x_1 + c_{22}^l x_2 + \cdots c_{2n}^l x_n) \\
& AND \cdots AND\ (y_m = c_{m0}^l + c_{m1}^l x_1 + c_{m2}^l x_2 + \cdots c_{mn}^l x_n),\ \ l = 1, 2, \cdots, M. \qquad (6.4)
\end{aligned}
$$

In general, the TSK model in Eq. (6.4) is equivalent to m MISO TSK FRM models such as:

$$
\begin{aligned}
R_k^l:\ & IF\ \ (x_1\ \ is\ \ X_1^{p_1})\ \ AND \cdots AND\ \ (x_n\ \ is\ \ X_n^{p_n}), \\
& THEN\ \ (y_k = c_{k0}^l + c_{k1}^l x_1 + c_{k2}^l x_2 + \cdots c_{kn}^l x_n),\ l = 1, 2, \cdots, M,\ k = 1, 2, \cdots, m. \qquad (6.5)
\end{aligned}
$$

Consider P input-output training data pairs:

$$(x_1(p), x_2(p), \cdots, x_n(p); y_1(p), y_2(p), \cdots, y_m(p)),\ \ p = 1, 2, \cdots, P. \qquad (6.6)$$

The fuzzy variables in Eq. (6.4) and (6.5) are the same as those in Eq. (6.1) except that in the l-th rule, the output variable y_k is the first-order linear polynomial function of inputs, and c_{ki}^{l} are the corresponding linear parameters x_i, $i = 0,1,2,\cdots,n$, $k = 1,2,\cdots,m$, $l = 1,2,\cdots,M$.

6.3.1 Linear parameter identification

The linear parameters c_{ki}^{l} $i = 0,1,2,\cdots,n$, $l = 1,2,\cdots,M$, in the consequent part of the fuzzy system in Eq. (6.5) can be updated similar to the centers in Eq. (6.2) by using some classical optimization algorithm such as the LSE [155]. Firstly, based the Eq. (6.2)-(6.3), the estimated output for the TSK FRM model in Eq. (6.5) can be determined by

$$\hat{y}_k = \hat{\mathrm{c}}_{\mathrm{k}0} + \hat{c}_{k1}x_1 + \hat{c}_{k2}x_2 + \cdots \hat{c}_{kn}x_n, \tag{6.7}$$

where each linear coefficient c_{ki}, $i = 0,1,2,\cdots,n$, $l = 1,2,\cdots,M$, is estimated by the following function:

$$\hat{c}_{ki} = f_{FRM}^{ki}(x_1, x_2, \cdots, x_n) = \overline{R}_{x_k} C_{ki}, \tag{6.8}$$

where $\overline{R}_{x_k}(t) = \dfrac{R_E^T(x, y_k)}{H_{1\times M} R_E(x, y_k)}$, $H^T{}_{1\times M} = (1, 1, \cdots, \ 1)^T \in \Re^{1\times M}$, $C_{ki} = (c_{ki}^1, \cdots, c_{ki}^M)^T$,

$i = 0,1,2,\cdots,n$, $R_E(x, y_k) = V_{F_j}(\hat{y}_k) == M_R \rhd V_E(x)$,

$V_E(x) = V_{E_1}(x_1) \rhd V_{E_2}(x_2) \rhd \cdots \rhd V_{E_n}(x_n)$, $x = (x_1, x_2, \cdots, x_n)$, $k = 1, 2, \ \cdots, m$.

It is seen from Eq. (6.7) that a fuzzy system with an FRM model is a nonlinear mapping from the input space $x \in \Re^n$ to the output space $f_{FRM}^{ki}(x) \in \Re$, represented by the product of two matrices. In general, it is difficult to find the exact solutions of Eq. (6.7). Therefore, in implementation, a hybrid algorithm will be proposed in this section to search for the estimated value $\hat{c}_{ki}$ of c_{ki}, from the following objective function:

$$\min_{c_{ki}} E(c_{ki}) = \min_{c_{ki}} e^T e = \min_{c_{ki}} \left\| f_{FRM}^{ki} - c_{ki}(p) \right\|_2 = \min_{c_{ki}} \left\| \overline{R}_{x_k} c_{ki} - c_{ki}^P \right\|_2 = \min_{c_{ki}} \sum_{p=1}^{P} (\overline{R}_{x_k} c_{ki} - c_{ki\,p})^2, \tag{6.9}$$

where $e = \overline{R}_{x_k} c_{ki} - c_{ki}^P$ is the error vector produced by c_{ki} and $c_{ki}^P = (c_{ki1}, c_{ki2}, \cdots, c_{kiP})^T$; $c_{ki}(0)$ is the initial parameter, $i = 0,1,2,\cdots,n$, $k = 1,2, \ \cdots, m$.

If $\overline{R}_{x_k}^T \overline{R}_{x_k}$ is nonsingular (realized by choosing suitable sampling data pairs), the linear parameters $c_{ki}^1, \cdots, c_{ki}^M$ in $f_{FRM}^k(x)$ can be trained by the recursive LSE [155]. The optimal estimation value $c_{ki}(p)$ can be determined by:

$$\begin{cases} c_{ki}(p) = c_{ki}(p-1) + K_{ki}(p)[u_0^p - b^T(x_0^p)c_{ki}(p-1)] \\ K_{ki}(p) = P_{ki}(p-1)b(x_0^p)[b^T(x_0^p)P_{ki}(p-1)b(x_0^p)+1]^{-1} \\ P_{ki}(p) = P_{ki}(p-1) - P_{ki}(p-1)b(x_0^p)[b^T(x_0^p)P_{ki}(p-1)b(x_0^p)+1]^{-1}b^T(x_0^p)P_{ki}(p-1)], \end{cases} \tag{6.10}$$

where $P_{ki}(0) = \sigma I$, σ is a large constant, and I is an identity matrix.

6.3.2 Nonlinear parameter identification

The nonlinear parameters in $V_{E_i}(x_i)$ of the NF-FRM model (i.e., the MF parameters) can be trained by using some nonlinear optimization algorithm such as the recursive RLM [156, 157]. Consider MISO fuzzy systems in Eq. (6.5) with input variables $x = (x_1, x_2, \cdots, x_n)$. If a Gaussian function is used as the MF of the input fuzzy variable x_i:

$$\mu_{X_i^l}(x_i) = \exp\left(\frac{-(x_i - a_{il})^2}{2\sigma_{il}^2}\right), \tag{6.11}$$

where a_{il} and σ_{il} are the respective center and spread of a Gaussian function, $l = 1, 2, \cdots, M$, $i = 1, 2, \cdots, n$.

$\theta_l = (a_{1l}, \sigma_{1l}, a_{2l}, \sigma_{2l}, \cdots, a_{nl}, \sigma_{nl}) \in \Re^{1\times 2n}$ is the nonlinear parameter to be updated by using the RLM. As an adaptive optimization algorithm, RLM possesses quadratic convergence even if the initial estimates are relatively poor. Additionally, the RLM algorithm has been proven globally convergent in many applications by properly choosing the step variables [157].

In the implementation, the firing strength ω_l of the l-th rule of the fuzzy system in Eq. (6.5) will be:

$$\omega_l = \bigcap_{i=1}^{n} \mu_{X_i^l}(x_i) = \bigcap_{i=1}^{n} e^{-(x_i - a_{il})^2 / 2\sigma_{il}^2} , \tag{6.12}$$

where $\bigcap$ can be chosen as a product *T*-norm operator. From Eq. (6.8), the *k*-th output variable of the FRM system will be

$$\hat{c}_{ki} = f_{FRM}^{ki}(x) = \overline{R}_{x_k} C_{ki} = \frac{\sum_{l=1}^{M} c_{ki}^{l} \cdot \omega_l}{\sum_{l=1}^{M} \omega_l} = \frac{a^{ki}}{b}, \tag{6.13}$$

where $a^{ki} = \sum_{l=1}^{M} c_{ki}^{l} \cdot \omega_l$; $b = \sum_{l=1}^{M} \omega_l$; and ω_l is the firing strength of the *l*-th fuzzy rule in Eq. (6.5), $l = 1, 2, \cdots, M$.

The parameters θ_l to be trained in the objective function in Eq. (6.9) can be expanded by a Tayler series [156]:

$$\begin{aligned} \theta_l(p+1) &\approx \theta_l(p) + \lambda_p (J_p^T J_p + \eta_p I)^{-1} J_p^T r_p \\ &= \theta_l(p) + \lambda_p H_p^{-1} J_p^T r_p \\ &= \theta_l(p) + (1-\alpha_p) H_p^{-1} J_p^T r_p, \end{aligned} \tag{6.14}$$

where $J_p \in \Re^{M \times 2n}$ is the Jacobin matrix, $H_p \in \Re^{2n \times 2n}$ is the modified Hessian matrix, $I_p \in \Re^{2n \times 2n}$ is the identity matrix, $\alpha_p = 1 - \lambda_p$ is the forgetting factor, and η_p is the learning factor.

The Hessian matrix can be described as

$$H_p = \alpha_p H_{p-1} + (1-\alpha_p)(J_p^T J_p + \eta_p I) . \tag{6.15}$$

The RLM algorithm can be derived by incorporating a regularization term in the Gauss-Newton optimization. In implementation, instead of computing the $2n \times 2n$ matrix $\eta_p I$ at each step, it considers only one of the diagonal elements of $J_p^T J_p$ such as:

$$H_p = \alpha_p H_{p-1} + (1-\alpha_p)(J_p^T J_p + 2n\eta_p \Lambda) , \tag{6.16}$$

where $\Lambda \in \Re^{2n \times 2n}$ has only one nonzero element located at $p\{\mathrm{mod}(2n)+1\}$ diagonal position or

$$\Lambda_{ii} = \begin{cases} 1, & if \quad i = p\{\text{mod}(2n)+1\}, \quad and \quad t > 2n \\ 0, & otherwise. \end{cases} \tag{6.17}$$

Then Eq. (6.16) can be rewritten as

$$H_p = \alpha_p H_{p-1} + (1-\alpha_p)(UV^{-1}U^T), \tag{6.18}$$

where $U^T = \begin{bmatrix} & & & J_p^T & & & \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$, and $V^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 2n\eta_p \end{bmatrix}$.

The inversion of matrix H_p in Eq. (6.18) can be computed by using the formula

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}, \tag{6.19}$$

where $A = \alpha_p H_{p-1}$, $B = (1-\alpha_p)U$, $C = V^{-1}$, and $D = U^T$.

If A and $(C^{-1} + DA^{-1}B)$ are nonsingular, from Eq. (6.19) and Eq. (6.14), by some manipulation [156], the RLM algorithm can be represented as

$$\begin{aligned} \theta_{p+1} &= \theta_p + \Phi_p J_p r_p, \\ \Phi_p &= \frac{1}{\alpha_p}[\Phi_{p-1} - \Phi_{p-1}US^{-1}U^T\Phi_{p-1}], \\ S &= \alpha_p V + U^T\Phi_{p-1}U, \end{aligned} \tag{6.20}$$

where $\theta_0 = 0$; $S \in \Re^{2\times2}$ and its inverse matrix are in 2×2 and are easy to be implemented for real-time applications; Φ_p is the covariance matrix with initial condition $\Phi_0 = \rho I$ and ρ is a positive quantity.

6.3.3 Parameter training of the FRM model

From the above analysis, the parameter training procedure is implemented as follows:

Firstly, the MIMO FRM model can be established using the following *h*-th input-output training data pair,

$$\{x_1(h), x_2(h), \cdots, x_n(h); y_1(h), y_2(h), \cdots, y_m(h)\}, \; h = 1, 2, \cdots, H. \tag{6.21}$$

Secondly, determine the MFs of each input variable with initial values of $\theta_l(0) = (a_{1l}(0), \sigma_{1l}(0), a_{2l}(0), \sigma_{2l}(0), \cdots, a_{nl}(0), \sigma_{nl}(0))$ and $(c_{ki}^1(0), \cdots, c_{ki}^M(0))$. The FRM model $y_{ki}(0) = f_{FRM}^{ki}(x(0)) = \overline{R}_{x_{ki0}} C_{ki}$ can be formulated using Eq. (6.13), $i = 0, 1, 2, \cdots, n$.

Thirdly, the linear parameters $\{c_{ki}^l\}$ and nonlinear parameters $\{a_{il}, \sigma_{il}\}$ in the FRM system will be trained by minimizing the following error function:

$$e_h = \frac{1}{2} \sum_{h=1}^{H} \left[\hat{y}_k(h) - y_k(h)\right]^2, \ k = 1, 2, \cdots, m, \ h = 1, 2, \cdots, H, \tag{6.22}$$

where $\hat{y}_k(h) = \hat{c}_{k0} + \hat{c}_{k1} x_1(h) + \hat{c}_{k2} x_2(h) + \cdots + \hat{c}_{kn} x_n(h)$, $\hat{c}_{ki}$ is the estimated linear parameter in Section 6.3.1, $i = 0, 1, 2, \cdots, n$, $l = 1, 2, \cdots, M$.

The parameter training of the FRM model is implemented using the following steps:

1) Fix the nonlinear MF parameters in antecedents of rules; train the linear parameters $c_{ki}^1, \cdots, c_{ki}^M$ in $f_{FRM}^{ki}(x)$ by the recursive LSE algorithm [155] in Eq. (6.10).

2) Fix the linear parameters in consequents of fuzzy rules; train the nonlinear MF parameters $\theta_l = (a_{1l}, \sigma_{1l}, a_{2l}, \sigma_{2l}, \cdots, a_{nl}, \sigma_{nl})$ using the RLM algorithm [156] by Eq. (6.20).

3) Calculate the outputs of the FRM model: $f_{FRM}^{ki}(x) = \dfrac{a^{ki}(h)}{b(h)}$, where $b(h) = \sum_{l=1}^{M} \omega_l$ and $a^{ki}(h) = \sum_{l=1}^{M} c_{ki}^l(h) \cdot \omega_l$.

4) Repeat steps 1) - 3) until all of the training data pairs are inputted to the system. Terminate the training process when the training error e_h in Eq. (6.22) is less than a threshold (e.g., 10^{-5} in this case) or the number of training epochs has reached a threshold (e.g., 200 in this case).

6.4 Numerical Simulation

6.4.1 The FRM prediction model

Some simulation tests will be undertaken in this section to examine the effectiveness of the proposed STP-based FRM models. The simulation test is to use the FRM model to predict the future values of a variable in a data set from Mackey-Glass equation in Eq. (6.23), which is a benchmark system in this research area [158, 159]:

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t), \tag{6.23}$$

where τ is a time delay.

Consider a general n-input-m-output FRM model for s-steps-ahead prediction. For an input horizon $\{x(k), x(k-s), \cdots, x(k-(n-1)s)\}$, the output variables at the k-th time instant are the forecasted values of states $\{x(k+s), x(k+2s), \cdots, x(k+ms)\}$. Each output variable $\hat{x}(k+js)$, $j=1,2,\cdots,m$, can be predicted by the following fuzzy model F_{FRM} formulated using the STP FRM model:

$$\left(\hat{x}(k+s), \hat{x}(k+2s), \cdots, \hat{x}(k+ms)\right) = F_{FRM}\left(x(k), x(k-s), \cdots, x(k-(n-1)s\right), \tag{6.24}$$

where $k=1,2,3,\cdots$; $s=1,2,3,\cdots$; $m=1,2,3,\cdots$.

6.4.2 Direct FRM prediction modeling

There will be two strategies to formulate the MIMO FRM prediction models: direct and recurrent prediction modeling. In using the direct FRM prediction, each future state of $\{\hat{x}(k+s), \hat{x}(k+2s), \cdots, \hat{x}(k+ms)\}$ can be forecasted through the following m MISO FRM prediction models from $\{x(k), x(k-s), \cdots, x(k-(n-1)s)\}$:

$$\hat{x}(k+s) = f_{FRM}^{1}\left(x(k),\ \ x(k-s),\ \ \cdots,\ \ x(k-(n-1)s)\right),$$

$$\hat{x}(k+2s) = f_{FRM}^{2}\left(x(k),\ \ x(k-s),\ \ \cdots,\ \ x(k-(n-1)s)\right),$$

$$\vdots$$

$$\hat{x}(k+ms) = f_{FRM}^{m}\left(x(k),\ \ x(k-s),\ \ \cdots,\ \ x(k-(n-1)s)\right). \tag{6.25}$$

Then the s-steps-ahead FRM prediction model will be $F_{FRM} = [f_{FRM}^{1}, f_{FRM}^{2}, \cdots, f_{FRM}^{m}]^{T}$. After the model parameters are identified by training, the model can be used to predict future states of $\{\hat{x}(k+s), \hat{x}(k+2s), \cdots, \hat{x}(k+ms)\}$.

6.4.3 The recurrent FRM prediction modeling

In using the recurrent FRM prediction, some of the predicted values will be fed back to the prediction model as historical information. Different from the direct feedforward FRM prediction models, the recurrent MIMO FRM model is:

$$\begin{cases} \hat{x}(k+s) = f_{FRM}\left(x(k),\ \ x(k-s),\ \ \cdots,\ \ x(k-(n-1)s)\right) \\ \hat{x}(k+2s) = f_{FRM}\left(\hat{x}(k+s),\ \ x(k),\ \ \cdots,\ \ x(k-(n-2)s)\right) \\ \qquad\qquad \cdots\cdots \\ \hat{x}(k+ms) = f_{FRM}\left(\hat{x}(k+(m-1)s),\cdots,\hat{x}(k+s),\ \ x(k),\cdots,x(k-(n-m+1)s)\right). \end{cases} \tag{6.26}$$

Eq. (6.26) is a specific prediction model of F_{FRM} in Eq. (6.24) where each FRM output has the same structure but different input variables. Consequently, only the parameters in the first prediction model $\hat{x}(k+s)$ need to be identified, and the following $m-1$ values $\{\hat{x}(k+2s),\cdots,\hat{x}(k+ms)\}$ can be recursively predicted using the same FRM model f_{FRM}.

Without loss of generality, a simulation example with $m=2$, $n=3$, $s=3$, $\tau=6$, will be used for demonstration. Three inputs $\{x(k-6),x(k-3),x(k)\}$ are used by the recognized FRM model to predict $\{x(k+3),x(k+6)\}$. Comparison will be undertaken between the direct FRM modeling and the recurrent FRM modeling. The simulation is undertaken in Matlab R2016a.

6.4.4 Construction of FRM prediction models

The fuzzy model inputs $\{x_1,x_2,x_3\}$ are the states $\{x(k-6),x(k-3),x(k)\}$; the outputs $\{x_4,x_5\}$ are the predicted values of { $\hat{x}(k+3)$, $\hat{x}(k+6)$ }. Suppose each input variable has two fuzzy sets with Sigmoid MFs of Small (S_i) and Large (L_i), represented as:

$$\mu_{S_i}(x_i) = \frac{1}{1+\exp(-\sigma_{i1}(x_i - a_{i1}))},$$

$$\mu_{L_i}(x_i) = \frac{1}{1+\exp(-\sigma_{i2}(x_i - a_{i2}))}, \tag{6.27}$$

where a_{ij} and σ_{ij} are the respective centers and spreads of the Sigmoid MFs, i = 1, 2, 3, j = 1, 2.

As an example of Eq. (6.4), the first-order TSK fuzzy rules of the FRM prediction model can be formulated as:

R_r : $IF \quad (x_1\ is\ X_1^r)\ AND\ (x_2\ is\ X_2^r)\ AND\ (x_3\ is\ X_3^r)$,

$$THEN\ (x_4 = b_{r40} + b_{r41}x_1 + b_{r42}x_2 + b_{r43}x_3)\ AND\ (x_5 = b_{r50} + b_{r51}x_1 + b_{r52}x_2 + b_{r53}x_3), \quad (6.28)$$

where $X_i^r \in \{S_i, L_i\}$, $r = 1, 2, \cdots, 8$.

According to the NF-STP construction and FRM optimization algorithms in Sections 6.2 to 6.3, the resulting FRM model will have 64 linear parameters and 12 nonlinear MF parameters to be updated. If a "product" T-norm operator and a "maximum" S-norm operator are used, and the fuzzy STP is used for fuzzy implication in FRM, by defuzzification, the overall output $\hat{x}(k+3)$ and $\hat{x}(k+6)$ will be:

$$\hat{x}_4 = \frac{\sum_{r=1}^{8} \omega_r (b_{r40} + b_{r41}x_1 + b_{r42}x_2 + b_{r43}x_3)}{\sum_{r=1}^{8} \omega_r},$$

$$\hat{x}_5 = \frac{\sum_{r=1}^{8} \omega_r (b_{r50} + b_{r51}x_1 + b_{r52}x_2 + b_{r53}x_3)}{\sum_{r=1}^{8} \omega_r}, \quad (6.29)$$

where ω_r (n = 3) is the firing strength of rule R_r, $r = 1, 2, \cdots, 8$.

6.4.5 FRM parameter training

In NF-STP parameter training, the objective function will be formulated as:

$$E(p) = \frac{1}{2}\left[\hat{x}_4(p) - x_4(p)\right]^2 + \frac{1}{2}\left[\hat{x}_5(p) - x_5(p)\right]^2, \ p = 1, 2, \cdots, P. \quad (6.30)$$

The estimated outputs from the FRM prediction model are $\hat{x}_4$ and $\hat{x}_5$ in Eq. (6.29), and x_4 and x_5 are actual outputs. There are 64 linear parameters to be updated by using the recursive LSE in Eq. (6.10), and 12 nonlinear parameters to be optimized by using the RLM algorithm in Eq. (6.20).

6.4.6 Structure and parameter identification

In this simulation, $x(0) = 1.2$ is selected in Eq. (6.23). The learning rate $\alpha = 0.5$ is chosen (i.e., between 0 and 1). With $P = 2000$ and $Q = 200$, Fig. 6.2 shows the MFs of these three input variables before and after training.

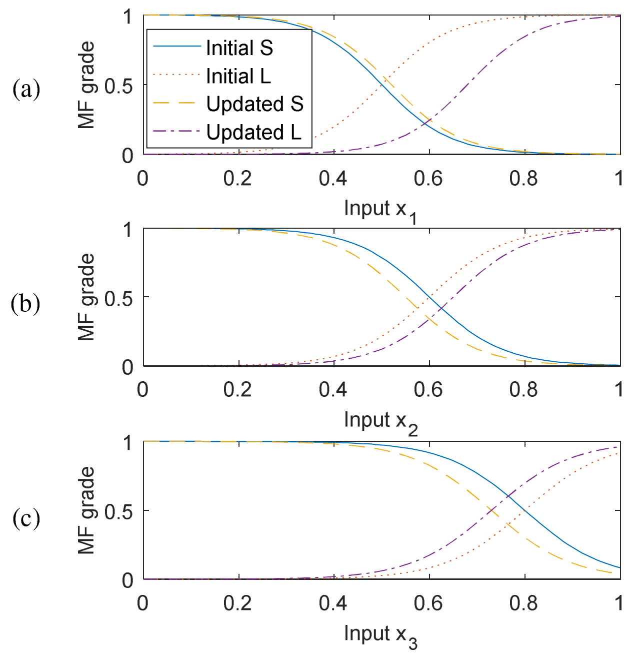
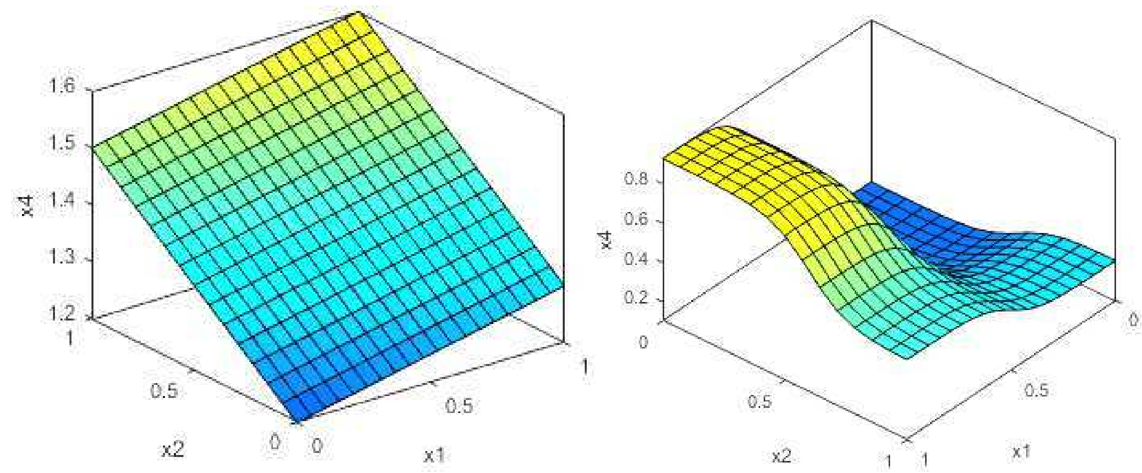


Fig. 6.2. MFs in prediction models before and after training: (a) Input variable x_1; (b) Input variable x_2; (c) input variable x_3.

In implementation, six groups of data pairs are used to train the FRM models. The mean results of these trained parameters in the six training groups will be used to recognize the recursive FRM model. After training, the mean errors between the actual outputs x_4 and predicted outputs $\hat{x}_4$ are 0.0108 and 0.0043 for the direct prediction and the recurrent prediction, respectively. The mean errors between x_5 and $\hat{x}_5$ are 0.0109 and 0.0045 for the direct prediction and the recurrent prediction, respectively. Thus the recurrent modeling outperforms the direct modeling, due to the use of historical information.

Next, the recurrent FRM prediction model will be used for advanced comparison analysis. Fig. 6.3 shows simulation results of estimated outputs. It is seen that the original linear fuzzy model before training becomes nonlinear after training, as the FRM applies nonlinear mapping from the input space to the output space in prediction modeling.



(a) (b)

Fig. 6.3. Decision surfaces of proposed recurrent FRM fuzzy prediction models: (a) Before training; (b) After training.

To further examine the effectiveness of the proposed recurrent FRM modeling, denoted as NF-STP, two other related techniques will be used for comparison for each output variable, because the outputs of the MIMO FRM model should be realized separately in simulation tests.

1) cFRM: the classical FRM technique without training. The cFRM predictor has the same reasoning structure and initial parameters as in the FRM models in NF-STP, or with 8 fuzzy rules in the 3-inputs and 2-outputs FRM model and 2 MFs for each input variable. It is used to compare the effectiveness of the training approach in NF-STP.

2) ANFIS: The ANFIS is a well-accepted NF model [159]. For comparison, ANFIS will have the same first-order TSK reasoning architecture as illustrated in Eq. (6.28). It has 8 rules, with 64 linear parameters and 12 nonlinear MF parameters, as in the NF-FRM model. The ANFIS is trained by using the same training algorithms as in the NF-STP: the nonlinear MF parameters are trained using the RLM algorithm and the consequent linear parameters are optimized using the recursive LSE. Additionally, both

the ANFIS and the NF-STP use the same initial parameters and learning rate, as well as the same training data and training epochs.

Fig. 6.4 shows the comparison results of the related predictors for x_4 and x_5, respectively. The root mean square errors for x_4 are 0.006, 0.021 and 0.069 for the NF-STP, ANFIS and cFRM, respectively. The root mean square errors for x_5 are 0.007, 0.023 and 0.073 for the NF-STP, ANFIS and cFRM, respectively.

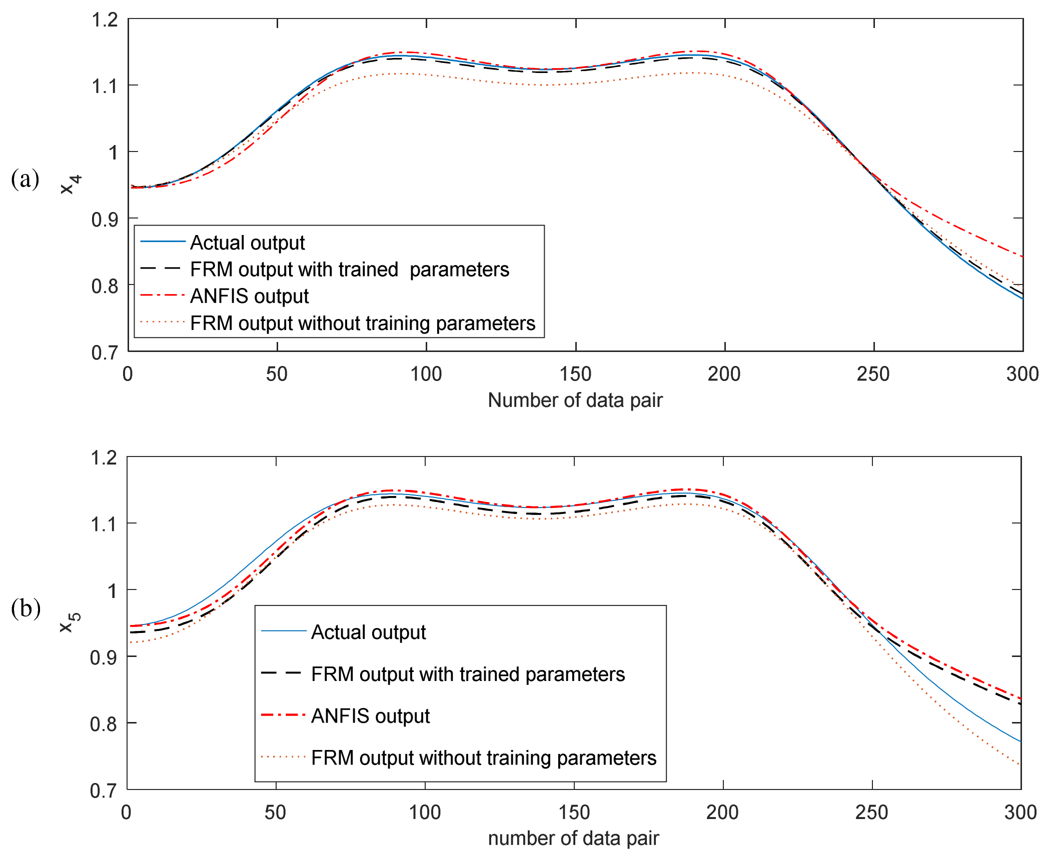


Fig. 6.4. Prediction results of the related predictors: (a) Output variable x_4; (b) Output variable x_5.

Hence, the proposed NF-STP outperforms the cFRM due to its more efficient system identification strategy and training operations. The NF-STP performs better than the classical ANFIS predictor due to its more effective modeling strategy. Therefore, the proposed NF-STP model can not only keep the merits of fuzzy systems in modeling transparent and stability, but also possess the adaptive capability in parameter training.

6.5 Concluding Remarks

Based on fuzzy STP, in this chapter, a new NF-STP technique has been proposed for FRM system identification and parameter optimization. In fuzzy structure identification, the fuzzy input-output relationship is modeled in a matrix representation while the fuzzy reasoning operations are realized by STP operations. Consequently, it can avoid the complex decomposition of multiple output variables, which still remains a challenge in the conventional structure identification of MIMO fuzzy systems. An NF-STP framework has been formulated to train FRM parameters. The effectiveness of the proposed NF-STP technique is verified by the simulation tests of prediction for the Mackey-Glass time series. The related contribution in this chapter has resulted in the following publications (accepted or submitted):

(1) H. Lyu, W. Wang, X. Liu, "Parameter identification and optimization of continuous MIMO fuzzy control systems by semi-tensor product", *Fuzzy Sets and Systems*, ISSN 0165-0114, in press, available online 14 June 2021.

(2) H. Lyu, W. Wang, X. Liu, "Neural-fuzzy model based on hierarchical structure matrix of MIMO systems via semi-tensor product", *IEEE Transactions on Fuzzy Systems*, submitted in 2021.

Chapter 7 Development of an Adaptive FRM Control System

It is well known that adaptive control systems are more suitable for applications in the plants with uncertainties or unknown parameters than those based on general mathematical models. Compared with traditional fuzzy control systems, an adaptive fuzzy system can adjust unknown parameters by some adaptive law. On the other hand, based on discussions in the previous chapters, the FRM models and fuzzy logic STP can keep similar properties as the conventional fuzzy logic models, while the matrix expression can provide advanced analysis of system properties. By this motivation, in this chapter, a novel adaptive fuzzy controller will be developed based on the FRM models and fuzzy STP approach for improving the control performance. The main contribution includes: 1) Formulating the structure of a closed-loop fuzzy control system based on the FRM and STP operations of logic matrices; 2) Designing an adaptive FRM controller in which system parameters can be adjusted; 3) Analyzing the stability of the closed-loop adaptive FRM control system based on Lyapunov methods.

This chapter is organized as follows. In Section 7.1, a closed-loop fuzzy control system is developed based on the FRM model and STP algorithm. An adaptive fuzzy control structure is developed based on FRM models in Section 7.2. An indirect adaptive FRM controller is designed in Section 7.3, while its properties are discussed in Section 7.4. Finally, the effectiveness of the proposed adaptive FRM control system is verified through simulations in Section 7.5.

7.1 Overview of Fuzzy Control based on the FRM and the STP

As discussed in Introduction, although the fuzzy control has some clear merits over traditional control methods, it still faces some challenges in applications. For example, the linguistic fuzzy inference could be difficult to model mathematically, or it lacks appropriate mathematical tools for advanced analysis. The current research works in this field mainly focus on the SISO, two-input and single-output (TISO), and adaptive fuzzy control systems [160, 161]; in contrast, MIMO fuzzy controllers are seldom considered because of their complexity in structures and nonlinearity [162]. This section focuses on the design of a non-adaptive FRM control system.

Consider a classical closed-loop feedback fuzzy control system shown in Fig. 7.1 [163], where the FRM model f_{FRM} is a general MIMO non-adaptive fuzzy controller. If the setpoint is *r* and the output of the process is *y*, the input signal of the FRM controller is the error variable $e = r - y$. If the fuzzy control is $u = f_{FRM}(e)$, mathematically f_{FRM} is used to represent a fuzzy relation between input and output variables: $E_1 \times E_2 \times \cdots \times E_n \to U_1 \times U_2 \times \cdots \times U_m$, which is a nonlinear mapping from $e = (e_1, e_2, \cdots, e_n)$ to $u = (u_1, u_2, \cdots, u_m)$. The respective universes of discourse of $e = (e_1, e_2, \cdots, e_n)$ and $u = (u_1, u_2, \cdots, u_m)$ are $E = E_1 \times E_2 \times \cdots \times E_n$ and $U = U_1 \times U_2 \times \cdots \times U_m$, where *n* and *m* are the respective numbers of inputs and outputs.

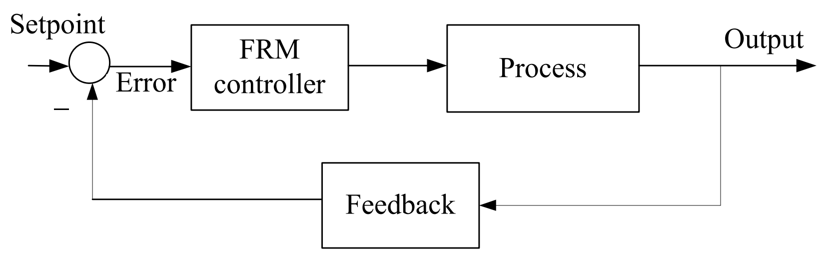


Fig. 7.1. The structure of the closed-loop FRM control system.

The design of a non-adaptive FRM control system is undertaken by using the FRM to replace the rule-base of a conventional fuzzy controller, and the fuzzy reasoning process is replaced by a fuzzy STP algorithm. Thus, the construction of the closed-loop FRM control system will focus on the matrix expression of fuzzy rules and fuzzy logic reasoning based on FRM models and STP algorithms.

Given a set of input-output training data pairs, the FRM model $M_R(e, u) = M_R(e_1, e_2, \cdots, e_n; u_1, u_2, \cdots, u_m)$ can be constructed according to the procedure from Eq. (4.1) to Eq. (4.11) in Section 4.2. Once an FRM model $M_R(e, u)$ is constructed, it is placed to the closed-loop system as the fuzzy controller. When the instant error $e(t)$

is inputted to the FRM controller, the fuzzy control signal $u(t)$ can be obtained. If setpoints are given, the actual errors $e^* = (e_1^*, e_2^*, \cdots, e_n^*)$ can be calculated using measurements from the system outputs. Then, the fuzzy output vector $u^* = (u_1^*, u_2^*, \cdots, u_m^*)$ can be determined from $M_R(e, u)$ and e^* by fuzzy STP operation (instead of the traditional fuzzy logic rule reasoning):

$$V_U(u_1^*, u_2^*, \cdots, u_m^*) = M_R \triangleright V_E(e_1^*, e_2^*, \cdots, e_n^*), \tag{7.1}$$

where $V_E(e_1^*, e_2^*, \cdots, e_n^*)$ and $V_U(u_1^*, u_2^*, \cdots, u_m^*)$ are the fuzzy input vector and output vector in the FRM controller, respectively. By proper defuzzification, the fuzzy control output $u^* = (u_1^*, u_2^*, \cdots, u_m^*)$ can be obtained.

The detailed design for the adaptive FRM control system will be discussed in the following sections.

7.2 Fundamental Design of the Adaptive FRM Controller

Fig. 7.2 provides an overview of the configuration of a classical adaptive fuzzy control system. The reference model is used for the fuzzy control system to follow. The plant is assumed to contain unknown parameters. An adaptive fuzzy controller is constructed [158], based on a fuzzy system whose parameters θ can be adjusted by some adaptive law such that the plant output $y(t)$ can track the reference model output $y_m(t)$.

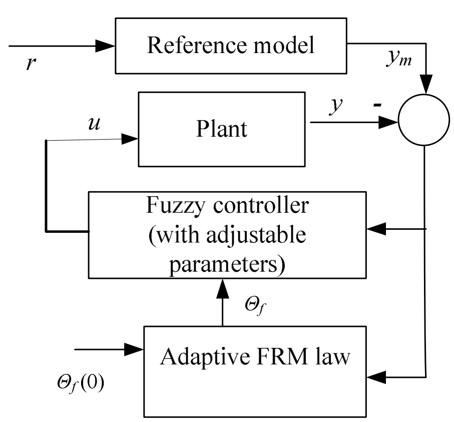


Fig. 7.2. Configuration of a classical adaptive fuzzy control system.

In general, adaptive fuzzy controllers can be classified into three categories [164]: indirect adaptive fuzzy control, direct adaptive fuzzy control, and hybrid (i.e., combined indirect and direct) adaptive fuzzy control [165]. In the proposed adaptive FRM control system, the fuzzy controller in Fig. 7.2 will be replaced by an FRM model. Correspondingly, it has three types FRM controllers: indirect adaptive FRM controller, direct adaptive FRM controller, and hybrid adaptive FRM controller. Therefore, a novelty of this work is related to development of an adaptive FRM control based on FRM models and fuzzy STP operations.

Assume that the plant is an *n*-th order continuous-time nonlinear system described by:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \quad \cdots \\ \dot{x}_n = f(x_1, x_2, \cdots, x_n) + g(x_1, x_2, \cdots, x_n)u \\ y = x_1 \end{cases} \tag{7.2}$$

where f and g are unknown functions, $u \in \Re$ and $y \in \Re$ are the input and output variables of the plant; $x = (x_1, x_2, \cdots, x_n)^T = (y, \dot{y}, \cdots, y^{(n-1)})^T \in \Re^n$ is the state vector of the plant.

In order for Eq. (7.2) to be controllable, it assumes that x is measurable and $g(x) > 0$ for x in its controllability region $U_C \in \Re^n$. In the nonlinear control literature [24], this system is in a normal form and has a relative degree of n.

Firstly, the nonlinear functions $f(x)$ and $g(x)$ in the plant are assumed to be unknown. The FRM models $\hat{f}(x)$ and $\hat{g}(x)$ will be designed to describe the input-output behaviors of $f(x)$ and $g(x)$, respectively. It is also assumed that some parameters θ in $\hat{f}(x)$ and $\hat{g}(x)$ can be modified during the online training operation. The control objective is to design a feedback adaptive fuzzy controller $u = u(x \mid \theta)$ and an adaptive law to adjust parameters θ. The adaptive law can be used to determine an adjusting θ so as to minimize tracking errors and the parameter estimation errors.

7.3 Design of the Indirect Adaptive Fuzzy Controller

In this section, the details will be discussed for designing an indirect adaptive fuzzy controller using FRMs.

7.3.1 Construction of an indirect adaptive FRM controller

Suppose that the plant is an n-th order nonlinear system described by Eq. (7.2). The objective is to design an FRM controller $u = u(x|\theta)$ and a fuzzy adaptive law for the parameter θ adjustment. Assume that variables $x(t)$, $\theta(t)$, and $u = u(x|\theta)$ are uniformly bounded, or $\|x(t)\| \le M_x < \infty$, $\|\theta(t)\| \le M_\theta < \infty$, and $\|u(x|\theta)\| \le M_u < \infty$ for $t \ge 0$, where M_x, M_θ, and M_u are parameters depending on the controlled plant.

Since functions $f(x)$ and $g(x)$ in the plant are unknown nonlinear functions, the following set of fuzzy rules will be used to describe the system behavior. Specifically $f(x)$ can be described by:

$$R_f^{l_f}: \ IF\ x_1 \text{ is } X_1^{p_1}\ and \cdots and\ x_n\ is\ X_n^{p_n},\ THEN\ f\ is\ F^{l_f},\ l_f = 1, 2, \cdots, M_f. \quad (7.3)$$

$g(x)$ can be characterized by:

$$R_g^{l_g}: \ IF\ x_1\ is\ X_1^{q_1}\ and \cdots and\ x_n\ is\ X_n^{q_n},\ THEN\ g\ is\ G^{l_g},\ l_g = 1, 2, \cdots, M_g, \quad (7.4)$$

where the state x_i is the i-th input of the FRM model; $E_{x_i}^f = \{X_{if}^1, \cdots, X_{if}^{N_{if}}\}$ and $E_{x_i}^g = \{X_{ig}^1, \cdots, X_{ig}^{N_{ig}}\}$ represent the input fuzzy sets; $l_f = p_1 + \sum_{i=2}^{n}[(p_i - 1)\prod_{j=1}^{i-1} N_{jf}]$, $p_i \in \{1, 2, \cdots, N_{if}\}$; $l_g = q_1 + \sum_{i=2}^{n}[(q_i - 1)\prod_{j=1}^{i-1} N_{jg}]$, $q_i \in \{1, 2, \cdots, N_{ig}\}$, $i = 1, 2, \cdots, n$; F^{l_f} and G^{l_g} denote the output fuzzy sets of f and g, respectively.

There are $M_f = \prod_{i=1}^{n} N_{if}$ fuzzy rules for FRM models $M_R^f(x)$ in Eq. (7.3) and $M_g = \prod_{i=1}^{n} N_{ig}$ fuzzy rules for FRM $M_R^g(x)$ in Eq. (7.4), respectively. Suppose that output fuzzy sets F^{l_f} and G^{l_g} in Eqs. (7.3) and (7.4) are normal with the centers $c_f^{l_f}$ and $c_g^{l_g}$, respectively, $l_f = 1, 2, \cdots, M_f$, $l_g = 1, 2, \cdots, M_g$. Based on these fuzzy rules and Eq. (4.24), both $f(x)$ and $g(x)$ can be approximated by the following FRMs:

$$M_R^f(x): \quad \hat{f}(x) := f_{FRM}^{M_R^f}(x) = \overline{R}_f \theta_f = \theta_f^T \overline{R}_f^T, \tag{7.5}$$

$$M_R^g(x): \quad \hat{g}(x) := f_{FRM}^{M_R^g}(x) = \overline{R}_g \theta_g = \theta_g^T \overline{R}_g^T, \tag{7.6}$$

where $\theta_f = (c_f^1, \cdots, c_f^{M_f})^T \in \Re^{M_f \times 1}$, $\overline{R}_f = \dfrac{(R_E^f)^T}{H_{1\times M_f} R_E^f} \in \Re^{1\times M_f}$, $H_{1\times M_f} = (1 \quad 1 \quad \cdots \quad 1) \in \Re^{1\times M_f}$,

$R_E^f(x, \hat{f}) = V_F(\hat{f}) = M_R^f \triangleright V_{Ef}(x)$, $V_{Ef}(x) = V_{E_{1f}}(x_1) \triangleright V_{E_{2f}}(x_2) \triangleright \cdots \triangleright V_{E_{nf}}(x_n)$,

$V_{E_{if}}(x_i) = (\mu_{X_{if}^1}(x_i), \cdots, \mu_{X_{if}^{N_{if}}}(x_i))^T$;

$\theta_g = (c_g^1, \cdots, c_g^{M_g})^T \in \Re^{M_g \times 1}$, $\overline{R}_g = \dfrac{(R_E^g)^T}{H_{1\times M_g} R_E^g} \in \Re^{1\times M_g}$, $H_{1\times M_g} = (1 \quad 1 \quad \cdots \quad 1) \in \Re^{1\times M_g}$,

$R_E^g(x, \hat{g}) = V_G(\hat{g}) = M_R^g \triangleright V_{Eg}(x)$, $R_{Eg}(x) = V_{E_{1g}}(x_1) \triangleright V_{E_{2g}}(x_2) \triangleright \cdots \triangleright V_{E_{ng}}(x_n)$,

$V_{E_{ig}}(x_i) = (\mu_{X_{ig}^1}(x_i), \cdots, \mu_{X_{ig}^{N_{ig}}}(x_i))^T$, $i = 1, 2, \cdots, n$.

7.3.2 Design of an indirect adaptive FRM controller

Based on the above matrix expressions, if nonlinear functions $f(x)$ and $g(x)$ are known, the control $u(t)$ can be made to cancel the nonlinearity. Specifically, if

$$e = y_m - y, \quad E = (e, \dot{e}, \cdots, e^{(n-1)})^T \in \Re^n, \text{ and} \tag{7.7}$$

$$K = (k_n, k_{n-1}, \cdots, k_1)^T \in \Re^n, \tag{7.8}$$

then, all roots of the polynomial $s^n + k_1 s^{n-1} + \cdots + k_n$ are in the open left-half complex plane. Then the control law can be selected as

$$u^* = \frac{1}{g(x)}[-f(x) + y_m^{(n)} + K^T E]. \tag{7.9}$$

Substituting Eq. (7.9) into Eq. (7.2), the closed-loop system will be governed by

$$y^{(n)} - y_m^{(n)} = -e^{(n)} = K^T E .$$

And then,

$$e^{(n)} + K^T E = e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e = 0 . \tag{7.10}$$

It can be shown that $e(t) \to 0$ as $t \to \infty$. Hence, the plant output y converges to the ideal output y_m. However, since $f(x)$ and $g(x)$ are unknown, it has no ideal controller recognizable in Eq. (7.9).

Fortunately, the FRMs in Eqs. (7.5)-(7.6) can be used to describe the input-output behaviors of $f(x)$ and $g(x)$. Therefore, the proposed approach is to replace $f(x)$ and $g(x)$ in Eq. (7.2) by FRMs $\hat{f}(x)$ and $\hat{g}(x)$, respectively, which are constructed from the rules in Eqs. (7.3)-(7.4).

If $\theta = (\theta_f^T,\ \theta_g^T)^T$ are the free parameters in $\hat{f}(x)$ and $\hat{g}(x)$, $\theta_f \in \Re^{M_f \times 1}$ and $\theta_g \in \Re^{M_g \times 1}$, then $\hat{f}(x) = \hat{f}(x | \theta_f)$ and $\hat{g}(x) = \hat{g}(x | \theta_g)$ can be employed to denote $f(x)$ and $g(x)$ with free parameters. Replacing $f(x)$ and $g(x)$ in Eq. (7.2) by the FRM models $\hat{f}(x | \theta_f)$ and $\hat{g}(x | \theta_g)$, respectively, the following fuzzy controller can be obtained

$$u = u_I(x | \theta) = u_I(x | \theta_f, \theta_g) = \frac{1}{\hat{g}(x | \theta_g)} [-\hat{f}(x | \theta_f) + y_m^{(n)} + K^T E] , \tag{7.11}$$

which is referred to as an adaptive FRM controller. If $\hat{f}(x) = f(x)$ and $\hat{g}(x) = g(x)$, there exist no uncertainties in f and g, then the controller $u_I(x | \theta)$ will become the ideal controller u^* of Eq. (7.9).

The next task is to design an adaptive law for θ_f and θ_g optimization.

7.3.3 Design of an indirect adaptive FRM law

After some manipulations on Eq. (7.2) and Eqs. (7.9)-(7.10), the closed-loop dynamics of the fuzzy system can be obtained:

$$e^{(n)} = -K^T E + [\hat{f}(x | \theta_f) - f(x)] + [\hat{g}(x | \theta_g) - g(x)]\, u_I . \tag{7.12}$$

Let

$$\Lambda = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & \cdots & \cdots & \cdots & -k_1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \tag{7.13}$$

Eq. (7.12) can be rewritten into the following vector form

$$\dot{E} = \Lambda E + b\{[\hat{f}(x|\theta_f) - f(x)] + [\hat{g}(x|\theta_g) - g(x)]u_I\}. \tag{7.14}$$

The optimal parameters can be recognized by

$$\theta_f^* = \arg\min_{\theta_f \in \Re^{M_f}} \left[\sup_{x \in \Re^n} \left\| \hat{f}(x|\theta_f) - f(x) \right\| \right], \tag{7.15}$$

$$\theta_g^* = \arg\min_{\theta_g \in \Re^{M_g}} \left[\sup_{x \in \Re^n} \left\| \hat{g}(x|\theta_g) - g(x) \right\| \right]. \tag{7.16}$$

Consequently, $\hat{f}(x|\theta_f^*)$ and $\hat{g}(x|\theta_g^*)$ can be the optimal values of $f(x)$ and $g(x)$, respectively [158]. Define the following minimum approximation error

$$w = [\hat{f}(x|\theta_f^*) - f(x)] + [\hat{g}(x|\theta_g^*) - g(x)]u_I. \tag{7.17}$$

Eq. (7.14) can be rewritten as

$$\dot{E} = \Lambda E + b\{[\hat{f}(x|\theta_f) - \hat{f}(x|\theta_f^*)] + [\hat{g}(x|\theta_g) - g(x|\theta_g^*)]u_I + w\}. \tag{7.18}$$

Substituting Eqs. (7.5)-(7.6) into Eq. (7.18), the following closed-loop dynamic equation can be obtained to specify the relationship between the tracking error e and the controller parameters θ_f and θ_g:

$$\dot{E} = \Lambda E + b[(\theta_f - \theta_f^*)^T \overline{R}_f^T + (\theta_g - \theta_g^*)^T \overline{R}_g^T u_I + w]. \tag{7.19}$$

The next task is to find an adaptive law to adjust FRM parameters θ_f and θ_g, so as to minimize the tracking error e and the parameter errors $e_f = \theta_f - \theta_f^*$ and $e_g = \theta_g - \theta_g^*$. Consider the Lyapunov function

$$V = \frac{1}{2}E^T PE + \frac{1}{2\gamma_f}(\theta_f - \theta_f^*)^T(\theta_f - \theta_f^*) + \frac{1}{2\gamma_g}(\theta_g - \theta_g^*)^T(\theta_g - \theta_g^*), \tag{7.20}$$

where $\gamma_f > 0$, $\gamma_g > 0$; P is a positive definite matrix that satisfies the following Lyapunov equation

$$\Lambda^T P + P^T \Lambda = -Q, \tag{7.21}$$

where $Q \in \Re^{n\times n}$ is a positive definite matrix. The time derivative of V along the closed-loop system trajectory Eq. (7.21) will be

$$\dot{V} = -\frac{1}{2}E^T PE + E^T Pbw + \frac{1}{\gamma_f}(\theta_f - \theta_f^*)^T(\dot{\theta}_f + \gamma_f E^T Pb\overline{R}_f^T) + \frac{1}{\gamma_g}(\theta_g - \theta_g^*)^T(\dot{\theta}_g + \gamma_g E^T Pb\overline{R}_g^T u_l) . \tag{7.22}$$

To minimize V, or equivalently to minimize the tracking error e, $\dot{V}$ should be negative. Since $-\frac{1}{2}E^T PE$ is negative, we can choose an adaptive law to minimize the approximation error w. Without loss of generality, we choose an adaptive law such that the last two terms in Eq. (7.22) are zero. Hence, the adaptive law will be

$$\dot{\theta}_f = -\gamma_f E^T Pb\overline{R}_f^T, \tag{7.23}$$

$$\dot{\theta}_g = -\gamma_g E^T Pb\overline{R}_g^T u_l, \tag{7.24}$$

which can be considered as a Lyapunov synthesis approach.

7.3.4 Design procedure of indirect adaptive FRM controllers

The procedures to design the indirect adaptive FRM controller are summarized as follows:

Step 1: Initialization:

- Specify $k_1, \cdots, k_n$, such that all roots of $s^n + k_1 s^{n-1} + \cdots + k_n = 0$ being located in the open left-half plane.
- Specify a positive definite $n \times n$ matrix *Q*.
- Solve Lyapunov equation Eq. (7.21) to obtain a symmetric positive definite matrix *P*.
- Determine the design constraints of parameters M_x, M_θ, M_u.

Step 2: Modeling of initial FRM controllers:

- Specify the universe of discourse $\tilde{E} = E_1 \times \cdots \times E_n$ for the state vector, and define N_{if} and N_{ig} fuzzy sets for $X_{if}^{p_i}$ and $X_{ig}^{q_i}$, respectively, here, $p_i \in \{1, 2, \cdots, N_{if}\}$, $q_i \in \{1, 2, \cdots, N_{ig}\}$, $i = 1, 2, \cdots, n$.
- Construct fuzzy rules of $\hat{f}(x|\theta_f)$ and $\hat{g}(x|\theta_g)$ in Eqs. (7.3)-(7.4), which comprise $N_{1f} \times N_{2f} \times \cdots \times N_{nf}$ rules as Eq. (7.3) and $N_{1g} \times N_{2g} \times \cdots \times N_{ng}$ rules as Eq. (7.4), respectively. Hence, the primary adaptive FRM controller is generated from Eqs. (7.5)-(7.6).
- Calculate the maximum values of $\mu_{F^{l_f}}$ and $\mu_{G^{l_g}}$. Input them into $\theta_f(0)$ and $\theta_g(0)$, to get $\hat{f}(x|\theta_f)$ and $\hat{g}(x|\theta_g)$ using Eqs. (7.5)-(7.6).

Step 3: Adaptive control processes:

- Import the feedback control law Eq. (7.11) to the plant Eq. (7.2), where $\hat{f}(x|\theta_f)$ and $\hat{g}(x|\theta_g)$ are obtained from Eqs. (7.5)-(7.6), where $K = (k_n, k_{n-1}, \cdots, k_1)^T$.
- Adjust the parameter vectors θ_f and θ_g online by the adaptive law Eqs. (7.23)-(7.24).

Fig. 7.3 illustrates the above design procedures of the proposed indirect adaptive fuzzy control system. In general, fuzzy IF-THEN rules in Eqs. (7.3)-(7.4) can be combined with the initial parameters $\theta_f(0)$ and $\theta_g(0)$ for modeling of an adaptive FRM model in the design of $\hat{f}(x|\theta_f)$ and $\hat{g}(x|\theta_g)$.

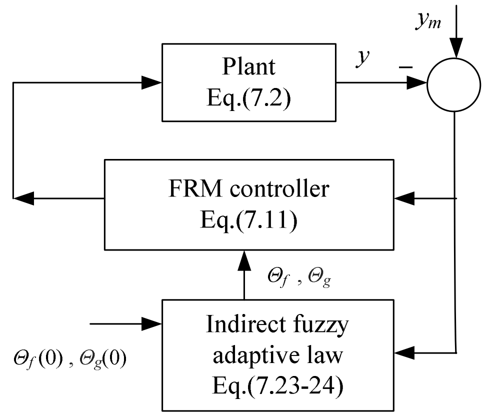


Fig. 7.3. The proposed indirect adaptive fuzzy control system.

7.4 Property Analysis of Indirect Adaptive FRM Control Systems

Consider the plant in Eqs. (7.3)-(7.4) with the controller $u_I(x|\,\theta_f,\theta_g)$ given in Eq. (7.11). If parameters θ_f and θ_g can be adjusted by the adaptive law in Eqs. (7.23)-(7.24), $\hat{f}(x|\theta_f)$ and $\hat{g}(x|\theta_g)$ in Eqs. (7.5)-(7.6) can be estimated online for functions $f(x)$ and $g(x)$ in Eq. (7.2). If $f(x)$ and $g(x)$ are bounded, the closed-loop fuzzy control system satisfies the following properties:

(1) All variables $x(t)$, $\theta(t)$ and $u = u(x|\,\theta)$ are uniformly bounded, or

$$\|\theta(t)\| \le M_\theta < \infty, \quad \|x(t)\| \le \|Y_m\| + \left(\frac{2V}{\lambda_{\min}}\right)^{1/2} \overset{\Delta}{=} M_x,$$

$$\|u(x|\theta)\| \le 2M_\theta + \frac{1}{b_L}[f^U + \|y_m^{(n)}\| + \|k\|\left(\frac{2V}{\lambda_{\min}}\right)^{1/2}] \overset{\Delta}{=} M_u < \infty,$$

where $t \ge 0$; $Y_m = \left(y_m, \dot{y}_m, \cdots, y_m^{(n-1)}\right)^T$; f^U is the supremum; $\lambda_{\min}$ is the minimum eigenvalue of P; and M_θ, M_x, M_u are parameters related to the controlled plant.

(2) If w is squared-integrable, then $\lim_{t\to\infty}\|e(t)\| = 0$, which can meet the requirements in designing a feedback FRM controller $u = u(x|\theta)$ and a fuzzy adaptive law for real-time adjustment of the parameter vector θ in Eq. (7.2).

The following remarks are given to summarize the adaptive fuzzy control systems:

Remark 1: The above properties (1) and (2) imply the original control objective, or the plant output y follows the ideal output y_m.

Remark 2: In real-world systems, there exist constraints for the state and control variables.

Remark 3: A real-world control problem could be much more complex than the plant model in Eq. (7.2). However, the proposed design strategy in this Chapter for the adaptive fuzzy control based on the FRM and fuzzy STP can be expanded for general modeling and control applications.

7.5 Applications and Simulation

The effectiveness of the proposed adaptive FRM control technique will be examined by some simulation tests in this section. As an example, consider a first-order inverted pendulum system [166]. The dynamic state equation of the inverted pendulum system with state variables (x_1, x_2) can be described as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{g\sin x_1 - \dfrac{mlx_2^2\cos x_1 \sin x_1}{m_c + m}}{l\left(\dfrac{4}{3} - \dfrac{m\cos^2 x_1}{m_c + m}\right)} + \dfrac{\dfrac{\cos x_1}{m_c + m}}{l\left(\dfrac{4}{3} - \dfrac{m\cos^2 x_1}{m_c + m}\right)}u \end{cases}, \tag{7.25}$$

where x_1 is the angle to the pole measured from the equilibrium position shown in Fig. 7.4; $x_2 = \dot{x}_1$; $g = 9.8m/s^2$ is the gravity acceleration; m is the mass of pole; m_c is the mass of cart; $2l$ is the length of pole; and u is the applied force.

It is seen that both $f(x)=\dfrac{g\sin x_1-\dfrac{mlx_2^2\cos x_1\sin x_1}{m_c+m}}{l\left(\dfrac{4}{3}-\dfrac{m\cos^2 x_1}{m_c+m}\right)}$ and $g(x)=\dfrac{\dfrac{\cos x_1}{m_c+m}}{l\left(\dfrac{4}{3}-\dfrac{m\cos^2 x_1}{m_c+m}\right)}$ are complex nonlinear functions of state (x_1, x_2). Then the FRM models $\hat{f}(x)$ and $\hat{g}(x)$ are constructed so as to implement the proposed adaptive FRM control system.

Before simulation tests, we assume $m=0.1$ kilogram, $m_c=1$ kilogram, and $l=0.5$ meters. The control objective is to make the output $y=x_1$, with the objective sinusoidal signal $y_m(t)=\dfrac{\pi}{30}\sin(t)$.

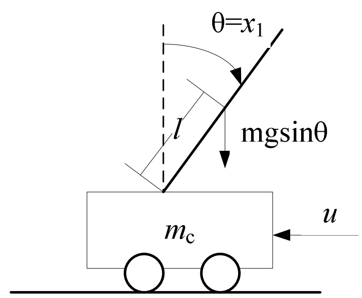


Fig. 7.4. An inverted pendulum system.

Before the proposed adaptive fuzzy control is applied, the primary offline design in Section 7.3.1 is used for construction of an indirect adaptive FRM controller. Assume that the universes of discourse of both x_1 and x_2 are $[-\pi,\pi]$. X_i^l is the fuzzy set with Gaussian MF $\mu_{X_i^l}(x_i)$ for the state variable x_i, $i=1,2$, $l=1,2,\cdots,5$.

Fig. 7.5 shows the corresponding MFs of x_1, similar procedures can be undertaken to optimize the MFs of x_2, where $\mu_{X_1^1}(x_1)=\exp\left(-\left(-\dfrac{x_1+\pi/6}{\pi/24}\right)^2\right)$,

$\mu_{X_1^2}(x_1)=\exp\left(-\left(\dfrac{x_1+\pi/12}{\pi/24}\right)^2\right)$, $\mu_{X_1^3}(x_1)=\exp\left(-\left(\dfrac{x_1}{\pi/24}\right)^2\right)$,

$$\mu_{X_1^4}(x_1) = \exp\left(-\left(\frac{x_1 - \pi/12}{\pi/24}\right)^2\right),\ \mu_{X_1^5}(x_1) = \exp\left(-\left(\frac{x_1 - \pi/6}{\pi/24}\right)^2\right).$$

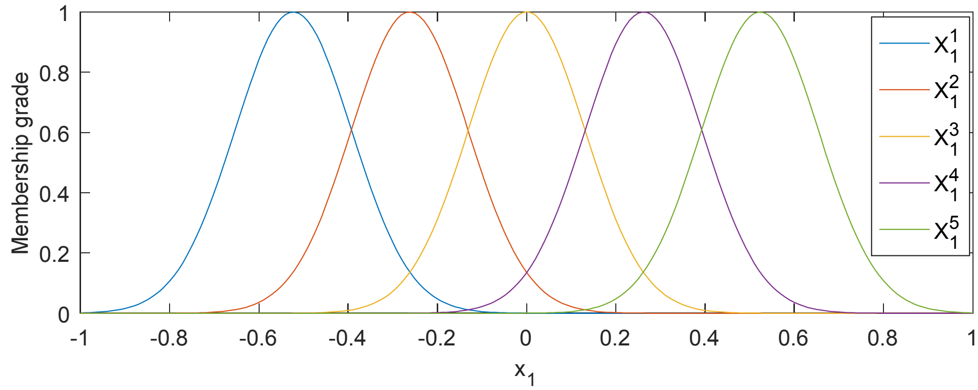


Fig. 7.5. The MFs of state variable of the inverted pendulum system.

According to the ranges of $f(x_1, x_2)$ and $g(x_1, x_2)$, select $\gamma_1 = 50$; $\gamma_2 = 1$; the equilibrium point $[x_1, x_2] = [0, 0]^T$; and the initial condition $[x_1(0), x_2(0)] = [-\pi/60, 0]^T$. If $k_1 = 2$, $k_2 = 1$, $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$, from Eq. (7.21), $P = \begin{bmatrix} 15 & 5 \\ 5 & 5 \end{bmatrix}$ can be obtained.

Suppose $u = 0$, then $f(x_1, x_2)$ equals the acceleration of angle x_1. The following initial fuzzy rules can be formulated from the given inverted pendulum system:

$$R_f^{(r)}:\ IF\ \ x_1\ \ is\ \ X_{1f}^r\ \ and\ \ x_2\ \ is\ \ X_{2f}^r,\ THEN\ \ f(x_1, x_2)\ \ is\ \ c_f^r,$$

where $X_{if}^r \in \{X_i^1, X_i^2, X_i^3, X_i^4, X_i^5\}$, and c_f^r is a singleton output for *f*, $i = 1, 2$, $r = 1, 2, \cdots, 25$.

Similarly, the fuzzy rules and FRM models of $g(x_1, x_2)$ can be constructed as:

$$R_g^{(r)}:\ IF\ \ x_1\ \ is\ \ X_{1g}^r\ \ and\ \ x_2\ \ is\ \ X_{2g}^r,\ THEN\ \ g(x_1, x_2)\ \ is\ \ c_g^r,$$

where $X_{ig}^r \in \{X_i^1, X_i^2, X_i^3, X_i^4, X_i^5\}$ and c_g^r is a singleton output for *g*, $i = 1, 2$, $r = 1, 2, \cdots, 25$.

The simulation of the proposed adaptive FRM control system can be run to track the ideal output y_m. Figs. 7.6 and 7.7 show the processing results. It is seen from Fig. 7.6

show that the state trajectory of x_1 and x_2, respectively. The results in Fig. 7.7 show that that the proposed adaptive FRM control system has the same control performance as the traditional adaptive fuzzy control (FLC) if the control structures and parameters are identical. Moreover, compared with non-adaptive control systems such as model prediction control (MPC), adaptive FRM control system can provide much better performance.

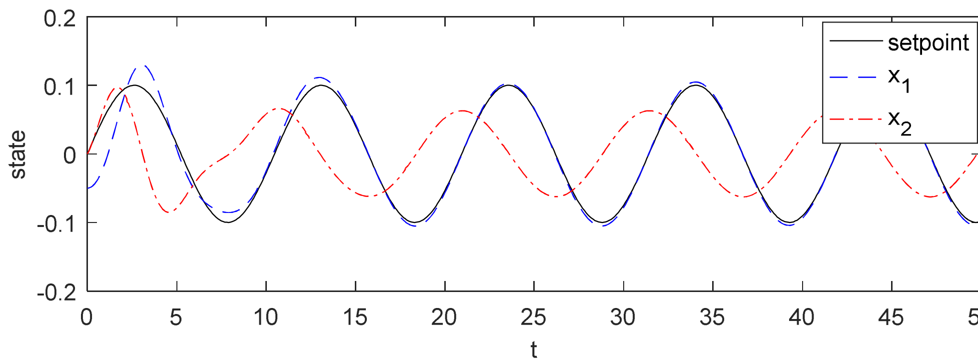


Fig. 7.6. Illustration of state variables x_1 and x_2 of the inverted pendulum system.

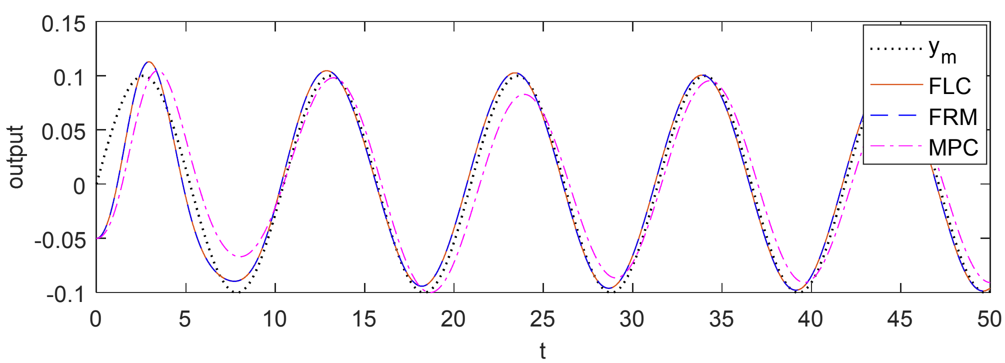


Fig. 7.7. Comparison of the state in the inverted pendulum system.

7.6 Concluding Remarks

In this chapter, a novel adaptive fuzzy control system has been developed on the basis of FRMs and fuzzy logic STP operations. An indirect adaptive FRM control law has been derived for MIMO nonlinear systems. The design procedures and algorithms of the indirect, direct and hybrid adaptive FRM control systems are similar. The effectiveness of the proposed indirect adaptive FRM control has been verified by simulation tests. The related contribution in this chapter has been resulted in the following paper:

(1) H. Lyu, W. Wang, X. Liu, "Indirect adaptive fuzzy control of nonlinear systems based on fuzzy relation matrix and semi-tensor product", *IEEE Transactions on Fuzzy Systems*, under revision, 2021.

Chapter 8 Conclusions and Future Works

8.1 Research Conclusions

For general multi-variable systems, it is usually difficult to derive precise mathematical models for MIMO nonlinear systems due to coupled variables, uncertainty, and unpredictable disturbance, etc. In this dissertation, we have explored new fuzzy formulation techniques based on the proposed FRM models and fuzzy logic STP algorithms. This work aims to improve fuzzy rule representation, fuzzy reasoning processes, fuzzy parameter optimization and adaptive fuzzy control systems, using FRM matrix expression. The main contributions are summarized as follows:

1) STP of multi-valued logic matrices is proposed for the matrix expression of multi-variable fuzzy relation based on some preliminaries about conventional STP operations and multi-dimensional data representation. The fundamental properties of STPs are also given (in Chapter 2).

2) The theoretical system of matrix expression is proposed for multi-variable fuzzy systems. The vector expression of fuzzy variables and fuzzy relations and matrix expression of multi-variable fuzzy relations are proposed. Then, the relation matrix of multiple fuzzy variables is developed for the MIMO fuzzy rules-based systems based on STP of logic matrices (in Chapter 3).

3) It is demonstrated that a general fuzzy system with an FRM model is a nonlinear mapping from input to output variables and it can be represented by the product of two matrices, after the fuzzy relation matrix (FRM) theory is extended from two-variable relation matrix to the conventional MIMO fuzzy systems based on matrix expression of multi-dimensional data and the fuzzy logic STP algorithms, and then, Next, the FRM models are identified by a direct modeling and an indirect identification strategy using sampling input-output training data (in Chapter 4).

4) A universal approximation is proposed for a nonlinear multi-variable function via FRM models and fuzzy logic STP algorithms. Then, the design algorithms are derived to model TISO, MISO and MIMO fuzzy systems, and the approximation accuracy is calculated for these fuzzy systems based on FRM models and fuzzy logic STP algorithms (in Chapter 5).

5) System identification and parameter training are proposed for FRM models based on a new neural-fuzzy STP network and fuzzy logic STP operations, where the linear and nonlinear parameters in FRM models can be updated by a hybrid optimization algorithm. In addition, the detailed parameter identification methods and training procedures are summarized (in Chapter 6).

6) A closed-loop fuzzy control system is constructed through FRM models and fuzzy logic STP. An indirect adaptive fuzzy controller is constructed on basis of FRM models for a n-order nonlinear system with unknown parameters, and then a feedback adaptive FRM controller and an adaptation law are designed for adjusting the unknown parameter. The tracking convergence is investigated by employing a Lyapunov function candidate (in Chapter 7).

The effectiveness of the related modeling and analysis techniques has been examined through some numerical simulation examples. Test results show that the proposed FRM is an efficient modeling technique for multiple-variable fuzzy models. Its identification approach can realize the matrix expression of the fuzzy systems and train parameters of MIMO FRM models. Additionally, the simulation results have shown that the fuzzy systems based on FRMs have better performance than other related modeling and identification techniques. It has potential to be implemented for real-world engineering applications in system control, system state prognostics, and pattern classification.

8.2 Future Works

While working on these research areas, we recognized that it is a new theoretical framework for MIMO fuzzy systems based on FRM models and fuzzy logic STP operations, so there are several potential topics to be explored in the future, specifically:

1) The current work mainly addressed the theoretical research about FRM models, and the tests were based on the numerical simulations. The future research will implement the FRM STP technology for real-world dynamic systems with noise and uncertainty, for applications such as system control, system state prognostics, and pattern classification.

2) The current numerical simulation examples were mainly based on some simple mathematical models with limited conditions. In the future work, real-world examples will be applied to investigate the effectiveness of the proposed theories and the potential for real engineering applications.

3) The FRM models were constructed mainly based on input-output training data pairs, which would be challenging in accurate and reliable representative data collection and implementation. Another future research theme will be related to investigate the complement of sampling database in FRM modeling and processing.

4) An FRM model in the current study is a matrix with each entry normalized within [0, 1], to fit the fundamental properties of conventional matrices. Advanced studies will be undertaken to propose new techniques in exploring the operations and properties of FRMs under quasi-singularity and redundancy conditions.

Appendix A
The proof of Eq. (4.29) in Section 4.3.2

In Section 4.3.2, the indirect modeling is discussed based on input-output training data in Eq. (4.21) under the condition that the parameters $C_j = (c_j^1, \cdots, c_j^M)^T$ are unknown, the FRM system is identified by using a LSE optimization algorithm in Eq. (4.22)-(4.29).

Given Q pairs of input-output training data sets:

$$\hat{C}_j = (\overline{R}_{x_q}^T \overline{R}_{x_q})^{-1} \overline{R}_{x_q}^T Y_j . \tag{4.29}$$

The proof of Eq. (4.29) is discussed as below.

To optimize Eq. (4.24) using the LSE, a general approach is to derive the derivative of $E(C_j)$ with respect to C_j and set it to zero.

With $C_j^T \overline{R}_{x_q}^T Y_j = Y_j^T \overline{R}_{x_q} C_j$, $E(C_j)$ in Eq. (4.25) can be expanded as

$$\begin{aligned} E(C_j) &= (C_j^T \overline{R}_{x_q}^T - Y_j^T)(\overline{R}_{x_q} C_j - Y_j) \\ &= C_j^T \overline{R}_{x_q}^T \overline{R}_{x_q} C_j - 2Y_j^T \overline{R}_{x_q} C_j + Y_j^T Y_j . \end{aligned} \tag{A.1}$$

Taking the partial derivative of $E(C_j)$ with respect to C_j yields

$$\frac{\partial E(C_j)}{\partial C_j} = 2\overline{R}_{x_q}^T \overline{R}_{x_q} C_j - 2\overline{R}_{x_q}^T Y_j . \tag{A.2}$$

Let

$$\frac{\partial E(C_j)}{\partial C_j} = 0 , \tag{A.3}$$

at $C_j = \hat{C}_j$. The following normal equation can be obtained:

$$\overline{R}_{x_q}^T \overline{R}_{x_q} \hat{C}_j = \overline{R}_{x_q}^T Y_j. \tag{A.4}$$

If $\overline{R}_{x_q}^T \overline{R}_{x_q}$ is nonsingular, then $\hat{C}_j$ can be solved uniquely as

$$\hat{C}_j = (\overline{R}_{x_q}^T \overline{R}_{x_q})^{-1} \overline{R}_{x_q}^T Y_j. \tag{A.5}$$

This is the proof of Eq. (4.29) in Chapter 4.

References

[1] E. Brusa, D. Ferretto, C. Ambra, *Systems Engineering and Its Application to Industrial Product Development*, 1st ed., Springer, 2018.

[2] K. Ogata, *Modern Control Engineering*, 5th Edition, Prentice Hall Edition, 2010.

[3] W. Chen, M. Saif, "Output feedback controller design for a class of MIMO nonlinear systems using high-order sliding-mode differentiators with application to a laboratory 3-D crane," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 11, pp. 3985-3997, 2008.

[4] N. A. Shirazee, A. Basak, "Elect ropermanent suspension system for acquiring large air-gaps to suspend loads," *IEEE Transaction on Magnetics*, vol. 31, no. 6, pp. 4193-4195, 1995.

[5] A. K. Tangirula, *Principles of System Identification: Theory and Practice*, CRC Press, Tayor and Francis Group, 2015.

[6] Y. Guo, D. Zhang, Z. Li, Q. wang. "Overviews on the applications of the Kuramoto model in modern power system analysis," *International Journal of Electrical Power and Energy Systems*, vol. 129, pp. 1-11, 2021.

[7] G. Goodwin, S. Graebe, M. Salgado, *Control System Design*, Prentice Hall, 2001.

[8] C. S. Chiu, "Derivative and integral terminal sliding mode control for a class of MIMO nonlinear systems," *Automatica*, vol. 48, no. 2, pp. 316-326, 2012.

[9] Q. Zhu, et al, "Review of rational nonlinear dynamic system modeling, identification, and control," *International Journal of Systems Science*, vol. 46, no. 12, pp. 2122-2133, 2015.

[10] J. J. Buckley, "Universal fuzzy controller," *Automatica*, vol. 28, no. 6, pp. 1245-1248, 1992.

[11] H. Ying, "Necessary conditions for some typical fuzzy systems as universal approximators," *Automatica*, vol. 33, no. 7, pp. 1333-1338, 1997.

[12] D. Cheng, H. Qi, Z. Li, *Analysis and Control of Boolean Networks- A Semi-tensor Product Approach*, 1st ed., Springer, pp. 1-98, 2011.

[13] H. Fan, J. Feng, M. Meng, B. Wang, "General decomposition of fuzzy relations: semi-tensor product approach," *Fuzzy Sets and Systems*, vol. 384, pp. 75-90, 2020.

[14] Y. Li, H. Li, W. Sun, “Event-triggered control for robust set stabilization of logical control networks,” *Automatica*, vol. 95, pp. 556-560, 2018.

[15] G. Giangiacomo, “Fuzzy logic programming and fuzzy control,” *Studia Logica*, vol. 79, no. 2, pp. 231-254, 2005.

[16] A. Ge, Y. Wang, Z. Liu, P. Jiang, “Hierarchical fuzzy control of multivariable systems via semi-tensor product method,” *Proceedings of the 33rd Chinese Control Conference*, Nanjing, China, pp. 4551-4556, 2014.

[17] J Feng, J Lam, Z Shu, Q Wang, “Internal positive preserved model reduction,” *International Journal of Control*, vol.83, no. 3, pp. 575-584, 2010.

[18] G. Boeing, “Visual analysis of nonlinear dynamical systems: chaos, fractals, self-similarity and the limits of prediction,” *Systems*, vol. 4, no. 4, pp. 37, 2016.

[19] J. Korenberg, M. Hunter, W. Ian, “The identification of nonlinear biological systems: volterra kernel approaches,” *Annals of Biomedical Engineering*, vol. 24, no. 2, pp. 250-268, 1996.

[20] V. Gintautas, “Resonant forcing of nonlinear systems of differential equations,” *Chaos*, vol. 18, no. 3, pp. 318-331, 2008.

[21] D. Canete, J. C. Galindo, I. Garcia, *System Engineering and Automation: An Interactive Educational Approach*, Berlin: Springer, 2011.

[22] K. Campbell, David, “Nonlinear physics: fresh breather,” *Nature*, vol. 432, no. 16, pp. 455-456, 2004.

[23] D. Lazard, “Thirty years of polynomial system solving, and now?” *Journal of Symbolic Computation*, vol. 44, no. 3, pp. 222-231, 2009.

[24] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, Wiley, 2013.

[25] D. Hinrichsen, A. J. Pritchard, *Mathematical Systems Theory I-Modelling, State Space Analysis, Stability and Robustness,* Springer Verlag, 2005.

[26] L. A. Zadeh, “Fuzzy sets,” *Information Control*, vol. 8, pp. 338-353, 1965.

[27] L. A. Zadeh, “Fuzzy algorithms,” *Information and Control*, vol. 12, no. 2, pp. 94-102, 1968.

[28] L. A. Zadeh, “Similarity relations and fuzzy ordering,” *Information Sciences*, vol. 3, no. 2, pp. 177-200, 1971.

[29] L. A. Zadeh, “Toward a theory of fuzzy systems,” *Aspects of Network and System Theory*, eds. R. E. Kalman and N. DeClaris, 1971.

[30] E. H. Mamdani, “Application for fuzzy algorithms for the control of a dynamic plant,” *Proceedings of the IEE*, vol. 121, pp. 1585-1588, 1974.

[31] A. Chibani, M. Chadli, P. Shi, N. B. Braiek, “Fuzzy fault detection filter design for T-S fuzzy systems in the finite-frequency domain,” *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1051-1061, 2017.

[32] L. X. Wang, *A Course in Fuzzy Systems and Control*, 1st ed., pp. 11-148, Prentice Hall PTR, Upper Saddle River, USA, 1997.

[33] L. A. Zadeh, “Outline of a new approach to the analysis of complex systems and decision processes,” *IEEE Transactions on Systems Man and Cybernetics*, vol. 3, no. 1, pp. 28-44, 1973.

[34] E. H. Mamdani, S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *International Journal of Man Machine Studies*, vol. 7, no. 1, pp. 1-13, 1975.

[35] L. A. Zadeh, “Fuzzy sets as a basis for a theory of possibility,” *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3-28, 1978.

[36] L. A. Zadeh, “Fuzzy logic, neural networks and soft computing,” *Communications of the ACM*, vol. 37, pp. 77-84, 1994.

[37] C. Li, J. Gao, J. Yi, G. Zhang, “Analysis and design of functionally weighted single-input-rule-modules connected fuzzy inference systems,” *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 56-71, 2018.

[38] K. H. Lee, “Robust adaptive fuzzy control by backstepping for a class of MIMO nonlinear systems,” *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 265-275, 2011.

[39] H. Li, Y. Wang, “A matrix approach to latticized linear programming with fuzzy-relation inequality constraints,” *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 4, pp. 781-788, 2013.

[40] G. Feng, *Analysis and Synthesis of Fuzzy Control Systems: A Model-based Approach*, 1st ed., CRC Press, Boca Raton, FL, USA, 2011.

[41] T. Li, S. Tong, G. Feng, “A novel robust adaptive-fuzzy-tracking control for a class of nonlinear multi-input/multi-output systems,” *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 150-160, 2010.

[42] Y. Liu et al., “A novel finite-time adaptive fuzzy tracking control scheme or nonstrict feedback systems,” *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 4, pp. 646-657, 2019.

[43] X. Hu, et al., “Robust sliding mode-based learning control for MIMO nonlinear nonminimum phase system in general form,” *IEEE Transactions on Cybernetics,* vol. 49, no. 10, pp. 3793-3805, 2019.

[44] X. Zhang, et al., “A novel three-dimensional fuzzy modeling method for nonlinear distributed parameter systems,” *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 489-501, 2019.

[45] O. Obajemu, et al., “A new fuzzy modeling framework for integrated risk prognosis and therapy of bladder cancer patients,” *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1565-1577, 2018.

[46] R. Langari, “Past, present and future of fuzzy control: a case for application of fuzzy logic in hierarchical control,” *18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS*, pp. 760-765, 1999.

[47] C. C. Lee, “Fuzzy logic in control system: fuzzy logic controller, Part I, II,” *IEEE Transactions on Systems Man and Cybernetics*, vol. 20, no. 2, pp. 404-435, 1990.

[48] G. M. Abdelnour, “Design of a fuzzy controller using input and output mapping factors,” *IEEE Transactions on Systems Man and Cybernetics*, vol. 21, no. 5, pp. 952-960, 1991.

[49] R. R. Yager, “Implementing fuzzy logic controller using a neural network,” *Fuzzy Sets and Systems*, vol. 48, no. 1, pp. 53-64, 1992.

[50] F. Bonslama, A. Ichiltawa, “Application of limit fuzzy controllers to stability analysis,” *Fuzzy Sets and Systems*, vol. 49, pp. 103-220, 1992.

[51] K. Hassan*, Nonlinear Systems,* Prentice Hall, 2001.

[52] H. J. Marquez, *Nonlinear Control Systems: Analysis and Design*, John Wiley & Sons Inc., 2002.

[53] H. K. Khalil, *Nonlinear Systems (3rd ed.)*, Upper Saddle River: Prentice Hall, 2002.

[54] X. Sun, Q. Zhang, G. Yang, "Admissibility analysis for interval type-2 fuzzy descriptor systems based on sliding model control," *IEEE Transactions on Cybernetic*, vol. 49, no. 8, pp. 2822-2834, 2019.

[55] P. Zhan, Z. Liu, H. Lyu, L. Zhao, P. Duan, G. Jia, "The design and application of a switch-air conditioner's zero static error fuzzy controller," *Transactions of the Chinese Society of Agricultural Engineering*, no. 10, pp. 219-224, 2013.

[56] X. Zhao, X. Wang, G. Zong et al., "Fuzzy-approximation-based adaptive output-feedback control for uncertain nonsmooth nonlinear systems," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3847-3859, 2018.

[57] Y. Li, Z. Gao, X. Liu, "An automatic fuzzy clustering segmentation algorithm with aid of set partitioning," *IEEE 15th International Conference on Industrial Informatics*, Emden, Germany, pp. 647-652, 2017.

[58] H. Huang, F. Meng, et al., "Brain image segmentation based on FCM clustering algorithm and rough set," *IEEE Access*, vol. 7, no. 1, pp. 12386-12396, 2019.

[59] K. Raymond, S. Gritzalis, J. Park, "Crytographic solutions for industrial internet of things: research challenges and opportunities," *IEEE Transactions on Industrial Information*, vol. 14, no. 8, pp. 3567-3569, 2018.

[60] B. Lei et al., "Walking imagery evaluation in brain computer interfaces via a multi-view multi-level deep polynomial network," *IEEE Transactions on Neural System and Rehabilitation Engineering*, vol. 27, no. 3, pp. 497-506, 2019.

[61] S. Bai, et al., "Automatic ensemble diffusion for 3D shape and image retrieval," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 88-101, 2019.

[62] M. Mohammadi et al., "Deep learning for IoT big data and streaming analytics: a survey," *IEEE Transactions on Communication Surveys and Tutorials*, vol. 20, no. 4, pp. 2923-2960, 2019.

[63] F. Celesti et al., "Why deep learning is changing the way to approach NGS data processing: a review," *IEEE Transactions on Reviews in Biomedical Engineering*, vol. 11, no. 2, pp. 68-76, 2018.

[64] J. Deutsch, D. He, “Using deep learning-based approach to predict remaining useful life of rotating components,” *IEEE Transactions on System Man Cybernetic, part C, Systems*, vol. 48, no. 1, pp. 11-20, 2018.

[65] L. Wang, *A Course in Fuzzy Systems*, Prentice Hall Press, USA, 1999.

[66] H. Long, “A note on the rates of uniform approximation of fuzzy systems,” *International Journal of Computer Integrated Systems*, vol. 4, no. 4, pp. 712-727, 2011.

[67] K. Zaras, “Rough approximation of a preference relation by a multi-attribute dominance for deterministic, stochastic and fuzzy decision problems,” *European Journal of Operational Research*, vol. 159, pp. 196-206, 2004.

[68] S. Cao, Y. Wang, J. Li, “Approximation of fuzzy neural networks based on choquet integral,” *Journal of Intelligent Fuzzy Systems*, vol. 31, pp. 691-698, 2016.

[69] A. Shakiba, et al, “An intuitionistic fuzzy approach to S-approximation spaces,” *Journal of Intelligent Fuzzy Systems*, vol. 30, pp. 3385-3397, 2016.

[70] H. Li, J. Wang, H. Du, H. Karimi, “Adaptive sliding mode control for Takagi-Sugeno fuzzy systems and its applications,” *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 531-542, 2018.

[71] G. Feng, “A survey on analysis and design of model-based fuzzy control systems,” *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 5, pp. 676-697, 2006.

[72] M. Jiang, X. Yuan, “A fuzzy inference modeling method for nonlinear systems by using triangular pyramid fuzzy system,” *Journal of Intelligent Fuzzy Systems,* vol. 33, pp. 1187-1196, 2017.

[73] C. Chak, G. Feng, J. Ma, “An adaptive fuzzy neural network for MIMO system model approximation in high-dimensional spaces,” *IEEE Transactions on Systems Man and Cybernetics part B, Cybernetic*, vol. 28, no. 3, pp. 436-446, 1998.

[74] U. Kula, B. Ocaktan, “A reinforcement learning algorithm with fuzzy approximation for semi Markov decision problems,” *Journal of Intelligent Fuzzy Systems,* vol. 28, pp. 1733-1744, 2015.

[75] S. Tong, Y. Li, P. Shi, “Fuzzy adaptive backstepping robust control for SISO nonlinear system with dynamic uncertainties,” *Information Sciences*, vol. 179, pp. 1319-1332, 2009.

[76] X. Liu, H. Wang, C. Gao, M. Chen, "Adaptive fuzzy funnel control for a class of strict feedback nonlinear systems," *Neurocomputing*, vol. 241, pp. 71-80, 2017.

[77] N. Wang, S. Tong, "Observer-based adaptive fuzzy control of a class of MIMO non-strict feedback nonlinear systems," *Journal of Frankin Institute,* vol. 355, pp. 4873-4896, 2018.

[78] B. Chen , X. Liu , K. Liu , C. Lin, "Adaptive control for nonlinear MIMO time-delay systems based on fuzzy approximation," *Information Sciences*, vol. 222, pp. 576-592, 2013.

[79] J. Dai, C. Chang, W. Xu and Z. Ye, "Linear precoder optimization for MIMO systems with joint power constraints," *IEEE Transactions on Communications*, vol. 60, no. 8, pp. 2240-2254, 2012.

[80] S. Lyashevskiy, "Constrained optimization and control of nonlinear systems: new results in optimal control," *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 1, pp. 541-546, 1996.

[81] Z. S. Xu, "Goal programming models for obtaining the priority vector of incomplete fuzzy preference relation," *International Journal of Approximate Reasoning*, vol. 36, pp. 261-270, 2004.

[82] G. Fu, "A fuzzy optimization method for multicriteria decision making: an application to reservoir flood control operation," *Expert Systems with Applications*, vol. 34, no. 1, pp. 145-149, 2008.

[83] C. Xu, "Fuzzy optimization of structures by the two-phase method," *Computers and Structures*, vol. 31, no. 4, pp. 575-580, 1989.

[84] P. Mertikopoulos, E. V. Belmega and A. L. Moustakas, "Matrix exponential learning: distributed optimization in MIMO systems," *2012 IEEE International Symposium on Information Theory Proceedings*, pp. 3028-3032, 2012.

[85] K. C. Lam, A. T. So, et al., "An integration of the fuzzy reasoning technique and the fuzzy optimization method in construction," *Project Management Decision-making*, vol. 19, no. 1, pp. 63-76, 2001.

[86] M. Dorigo, C. Blum, "Ant colony optimization theory: a survey," *Theoretical Computer Science*, vol. 344, no. 2, pp. 243-278, 2005.

[87] Yang, X. S, *Nature-inspired Metaheuristic Algorithms*, Frome, UK: Luniver Press, 2010.

[88] J. Lee, "On method for improving performance of PI-type fuzzy logic controllers," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 4, pp. 298-301, 1993.

[89] L. X. Wang, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems Man and Cybernetics*, vol. 22, no. 6, pp. 1414-1427, 1992.

[90] T. Jiang, Y. Li, "Multimode oriented polynomial transformation based defuzzification strategy and parameter learning procedure," *IEEE Transactions on Systems Man and Cybernetics*, vol. 27, no. 5, pp. 877-883, 1997.

[91] M. Chen, S. Ge, B. Ren, "Adaptive tracking control of uncertain MIMO nonlinear systems with input constraints," *Automatica*, vol. 47, no. 2, pp. 452-465, 2011.

[92] V. O. Bragin, V. I. Vagaitsev, N. V. Kuznetsov, G. A. Leonov, "Algorithms for finding hidden oscillations in nonlinear systems. the Aizerman and Kalman conjectures and Chua's circuits," *International Journal of Computer and Systems Sciences*, vol. 50, no. 5, pp. 511-543, 2011.

[93] J. Zhao, C. Lin, F. Chao, "Wavelet fuzzy brain emotional learning control system design for MIMO uncertain nonlinear systems," *Frontiers in Neuroscience*, vol. 12, no. 1, pp. 1-14, 2019.

[94] T. Naderi, D. Materassi, G. Innocenti, R. Genesio, "Revisiting Kalman and Aizerman conjectures via a graphical interpretation," *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp. 670-682, 2019.

[95] K. Pal, R. K. Mudi and N. R. Pal, "A new scheme for fuzzy rule-based system identification and its application to self-tuning fuzzy controllers," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 4, pp. 470-482, 2002.

[96] X. L. Xie and G. A. Beni, "Validity measure for fuzzy clustering," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 3, pp. 841-846, 1991.

[97] J. Cao, H. Liu, P. Li, D. J. Brown, "State of the art in vehicle active suspension adaptive control systems based on intelligent methodologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 392-405, 2008.

[98] S. R. Vaishnav and Z. J. Khan, "Performance of tuned PID controller and a new hybrid fuzzy PD+I controller," *World Journal of Modeling and Simulation*, vol. 6, no. 2, pp. 141-149, 2010.

[99] S. N. Vassilyev, Y. I. Kudinov, et al., "Intelligent control systems and fuzzy controllers. II. trained fuzzy controllers, fuzzy PID controllers," *Automatic Remote Control*, vol. 81, pp. 922-934, 2020.

[100] D. Matko, R. Karba, B. Zupancic, "Neuro-fuzzy identification models," *Industrial Technology 2000. Proceedings of IEEE International Conference on*, vol. 2, pp. 650-655, 2000.

[101] Y. Chien, W. Wang, Y. Leu, T. Lee, "Robust adaptive controller design for a class of uncertain nonlinear systems using online T-S fuzzy-neural modeling approach," *IEEE Transactions on Systems Man and Cybernetics Part B: Cybernetics*, vol. 41, no. 2, pp. 542-552, 2011.

[102] A. Mahasneh, M. Aljarrah, M. Rababah, *et al.*, "Application of hybrid neural fuzzy system (ANFIS) in food processing and technology," *Food Engineering Review,* vol. 8, pp. 351-366, 2016.

[103] S. S. Zhou, G. Feng, and C. B. Feng, "Robust control for a class of uncertain nonlinear systems: adaptive fuzzy approach based on backstepping," *Fuzzy Sets and Systems*, vol. 151, no. 1, pp. 1-20, 2005.

[104] A. Marco, M. Vera, et al., "Temperature control in a MISO greenhouse by inverting its fuzzy model," *Computers and Electronics in Agriculture*, vol. 124, pp. 168-174, 2016.

[105] Y. Yu, B. Wang, J. Feng, "Input observability of Boolean control networks," *Neurocomputing*, vol. 333, pp. 22-28, 2019.

[106] H. Li, Y. Wang, "A matrix approach to latticized linear programming with fuzzy-relation inequality constraints," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 4, pp. 781-788, 2013.

[107] A. Ge, "A new approach to controller design of stochastic fuzzy systems," *2013 IEEE International Conference on Information and Automation (ICIA)*, pp. 43-47, 2013.

[108] Y. Yu, M. Meng, J. Feng, P. Wang, "Stabilizability analysis and switching signals design of switched Boolean networks," *Nonlinear Analysis Hybrid System*, vol. 30, pp. 31-44, 2018.

[109] M. Meng, J. Lam, J. Feng, K. Cheung, "Stability and guaranteed cost analysis of time-triggered Boolean networks," *IEEE Transactions on Neural Network Learning System*, vol. 29, no. 8, pp. 3893-3899, 2018.

[110] Y. Yu, J. Feng, J. Pan, D. Cheng, "Block decoupling of Boolean control networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3129-3140, 2019.

[111] H. Li, L. Xie, Y. Wang, "Output regulation of Boolean control networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2993-2998, 2017.

[112] Y. Yu, J. Feng, B. Wang, P. Wang, "Sampled-data controllability and stabilizability of Boolean control networks: nonuniform sampling," *Journal of Frankin Institute*, vol. 335, no. 12, pp. 5324-5335, 2018.

[113] J. Feng, H. Lyu, D. Cheng, "Multiple fuzzy relation and its application to coupled fuzzy control," *Asian Journal of Control*, vol. 15, no. 5, pp. 1313-1324, 2013.

[114] H. Li, Y. Wang, "Lyapunov-based stability and construction of Lyapunov functions for Boolean networks," *SIAM Journal of Control Optimal*, vol. 55, no. 6, pp. 3437-3457, 2017.

[115] Y. Zheng, H. Li, X. Ding, Y. Liu, "Stabilization and set stabilization of delayed Boolean control networks based on trajectory stabilization," *Journal of Frankin Institute*, vol. 354, pp. 7812-7827, 2017.

[116] H. Li, Y. Wang, "Further results on feedback stabilization control design of Boolean control networks," *Automatica*, vol. 83, pp. 303-308, 2017.

[117] D. Cheng, J. Feng, H. Lyu, "Solving fuzzy relational equations via semi-tensor product," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 390-396, 2012.

[118] D. Cheng, "Semi-tensor product of matrices and its application to Morgan's problem," *Science China F-Information Science*, vol. 44, pp. 195-212, 2001.

[119] D. Cheng, J. Ma, Q. Lu, et al., "Quadratic form of stable sub-manifold for power systems," *International Journal of Robust Nonlinear Control*, vol. 14, pp. 773-788, 2004.

[120] D. Cheng, X. Hu, Y. Wang, “Non-regular feedback linearization of nonlinear systems via a normal form algorithm,” *Automatica*, vol. 40, pp. 439-447, 2004.

[121] D. Cheng, G. Yang, Z. Xi, “Nonlinear systems possessing linear symmetry,” *International Journal of Robust Nonlinear Control*, vol. 17, pp. 51-81, 2010.

[122] E. Fornasini, M. Valcher, “Recent developments in Boolean networks control,” *Journal of Control Decision*, vol. 3, pp. 1-18, 2016.

[123] D. Cheng, H. Qi, “State-space analysis of Boolean networks,” *IEEE Transactions on Neural Network*, vol. 21, pp. 584-594, 2010.

[124] D. Cheng, H. Qi, “A linear representation of dynamics of Boolean networks,” *IEEE Transactions on Automatic Control*, vol. 55, pp. 2251-2258, 2010.

[125] K. Zhang, L. Zhang, S. Mou, “An application of invertibility of Boolean control networks to the control of the mammalian cell cycle,” *IEEE Transactions on Computer Biology and Information*, vol. 14, pp. 225-229, 2017.

[126] Y. J. Sun, F. Liu, S. W. Mei, “Polynomial approximation of a nonlinear system and its application to power system (II): applications,” *Electrical Machine Control*, vol. 14, pp. 7-12, 2010.

[127] M. Xu, Y. Wang, “Conflict-free coloring problem with application to frequency assignment,” *Journal of Shandong University*, vol. 545, pp. 64-69, 2015.

[128] D. Cheng, F. He, H. Qi, et al., “Modeling, analysis and control of networked evolutionary games,” *IEEE Transactions on Automatic Control*, vol. 60, pp. 2402-2415, 2015.

[129] X. Han, Z. Chen, Z. Liu, et al., “Calculation of siphons and minimal siphons in petri-nets based on semi-tensor product of matrices,” *IEEE Transactions on Systems Man Cybernetics*, vol. 47, pp. 531-536, 2017.

[130] Z. Liu, Y. Wang, D. Cheng, “Nonsingularity of feedback shift registers,” *Automatica*, vol. 55, pp. 247-253, 2015.

[131] Y. H. Wu, T. Shen, “Policy iteration approach to control residual gas fraction in IC engines under the framework of stochastic logical dynamics,” *IEEE Transactions on Control Systems* and *Technology*, vol. 25, pp. 1100-1107, 2017.

[132] P. Duan, H. Lv, J. Feng, C. Liu, H. Li, "Indoor dynamic thermal control based on fuzzy relation model," *Control Theory and Application*, vol. 30, no. 2, pp. 215-221, 2013.

[133] H. Li, Y. Wang, "Boolean derivative calculation with application to fault detection of combinational circuits via the semi-tensor product method," *Automatica*, vol. 48, pp. 688-693, 2012.

[134] Y. Jia, X. Yang, "Optimization of control parameters based on genetic algorithms for spacecraft attitude tracking with input constraints," *Neurocomputing*, vol. 177, pp. 334-341, 2016.

[135] P. L. Guo, Y. Z. Wang, "Matrix expression and vaccination control for epidemic dynamics over dynamic networks," *Control Theory Technology*, vol. 14, pp. 39-48, 2016.

[136] P. Jiang, Y. Wang, A. Ge, "Multivariable fuzzy control based mobile robot odor source localization via semi-tensor product," *Math Problem and Engineering*, vol. 2015, pp. 720-736, 2015.

[137] D Cheng, H Qi, Y Zhao, *An Introduction to Semi-tensor Product of Matrices and Its Applications*, World Scientific, Singapore, 2012.

[138] H. Lyu, W. Wang, X. Liu, "Universal approximation of multi-variable fuzzy systems by semi-tensor product," *IEEE Transactions on Fuzzy Syst*ems, vol. 28, no. 11, pp. 2972-2981, 2020.

[139] C. Zhang, H. Lyu, P. Duan, Y. Song, "Fuzzy modeling of the semi-tensor product-based cold and hot electric installations," *Architecture Electric*, vol. 34, no. 9, pp. 59-64, 2015.

[140] H. Lyu, X. Hua, P. Duan, C. Zhang, "Air-conditioning system fuzzy control based on indoor dynamic thermal comfort," *Shandong Jianzhu Academic Journal*, vol. 29, no. 5, pp. 476-482, 2014.

[141] J. R. Jang, C. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Printice Hall, pp. 1-156, 1997.

[142] J. Maiers and Y. S. Sherif, "Applications of fuzzy set theory," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 175-189, 1985.

[143] G. Deschrijver, E. E. Kerre, "On the composition of intuitionistic fuzzy relations," *Fuzzy Sets and Systems*, vol. 136, no. 3, pp. 333-361, 2003.

[144] J. Loetamonphong, S. C. Fang, "Optimization of fuzzy relational equations with max-product composition," *Fuzzy Sets and Systems*, vol. 118, pp. 509-517, 2001.

[145] Y. Suh, "The performance of maximum likelihood and weighted least square mean and variance adjusted estimators in testing differential item functioning with nonnormal trait distributions," *Structural Equation Modeling: A Multidisciplinary Journal*, vol. 22, no. 4, pp. 568-580, 2015.

[146] R. E. Precup, H. Hellendoorn, "A survey on industrial applications of fuzzy control," *Computers in Industry*, vol. 62, no. 3, pp. 213-226, 2011.

[147] M. Hojati and S. Gazor, "Hybrid adaptive fuzzy identification and control of nonlinear systems," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 198-210, 2002.

[148] G. C. Mouzouris and J. M. Mendel, "Nonsingleton fuzzy logic systems: theory and application," *IEEE Transactions on Fuzzy Systems*, vol. 5, pp. 56-71, 1997.

[149] Z. Lu, L. Dong and J. Zhou, "Nonlinear least squares estimation for parameters of mixed weibull distributions by using particle swarm pptimization," *IEEE Access*, vol. 7, pp. 60545-60554, 2019.

[150] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 5, pp. 676-697, 2006.

[151] H. Li, J. Wang, H. Du, H. Karimi, "Adaptive sliding mode control for Takagi-Sugeno fuzzy systems and its applications," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 531-542, 2018.

[152] G. Wang, X. Li, "Mesh construction of PLF and its approximation process in Mamdani fuzzy system," *Journal of Intelligent Fuzzy Systems,* vol. 32, pp. 4213-4225, 2017.

[153] H. Ying, "A sufficient condition on a general class of interval type-2 Takagi-Sugeno fuzzy systems with linear rule consequent as universal approximators," *Journal of Intelligent Fuzzy Systems*, vol. 29, pp. 1219-1227, 2015.

[154] J. Zhang, Z. He, X. Wang and Y. Huang, "TSK fuzzy approach to channel estimation for MIMO-OFDM systems," *IEEE Signal Processing Letters*, vol. 14, no. 6, pp. 381-384, 2007.

[155] E. Lughofer, "Improving the robustness of recursive consequent parameters learning in evolving neuro-fuzzy systems," *Information Sciences*, vol. 545, pp. 555-574, 2021.

[156] D. Li, W. Wang, F. Ismail, "A fuzzy-filtered grey network technique for system state forecasting," *Soft Computing*, vol. 19, pp. 3497-3505, 2015.

[157] W. Wang, J. Vrbanek, "An evolving fuzzy predictor for industrial applications," *IEEE Transactions on Fuzzy Systems*, vol. 16, pp. 1439-1449, 2008.

[158] L. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Prentice-Hall, Englewood, Cliffs, NJ, 1994.

[159] J. R. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, pp. 353-360, 1997.

[160] F. M. F. Flaih, Lin Xiangning, Q. K. Mohsin and S. M. Dawoud, "Improving voltage profile in distribution system using PSO-TISO fuzzy in DSTATCOM control," *2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS)*, pp. 424-429, 2016.

[161] Z. Qu, Z. Du, Z. Liu, "Adaptive fuzzy control for SISO nonlinear time-delay systems," *2007 Chinese Control Conference*, pp. 246-249, 2007.

[162] H. Li, Y. Wang, "A matrix approach to latticized linear programming with fuzzy-relation inequality constraints," *IEEE Transactions on Fuzzy Systems*, vol. 21, pp. 781-788, 2013.

[163] A. Isidori, *Nonlinear Control Systems (3rd ed.)*, Berlin: Springer, 1995.

[164] Y. Liu, X. Liu, Y. Jing, Z. Zhang, "A novel finite-time adaptive fuzzy tracking control scheme or nonstrict feedback systems," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 4, pp. 646-657, 2019.

[165] L. Wang, "Stable adaptive fuzzy control of nonlinear systems," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 146-155, 1993.

[166] L. Wang, “Stable adaptive fuzzy controllers with application to inverted pendulum tracking,” *IEEE Transactions on Systems, Man, and Cybernetics: Cybernetics*, vol. 26, no. 5, pp. 677-691, 1996.