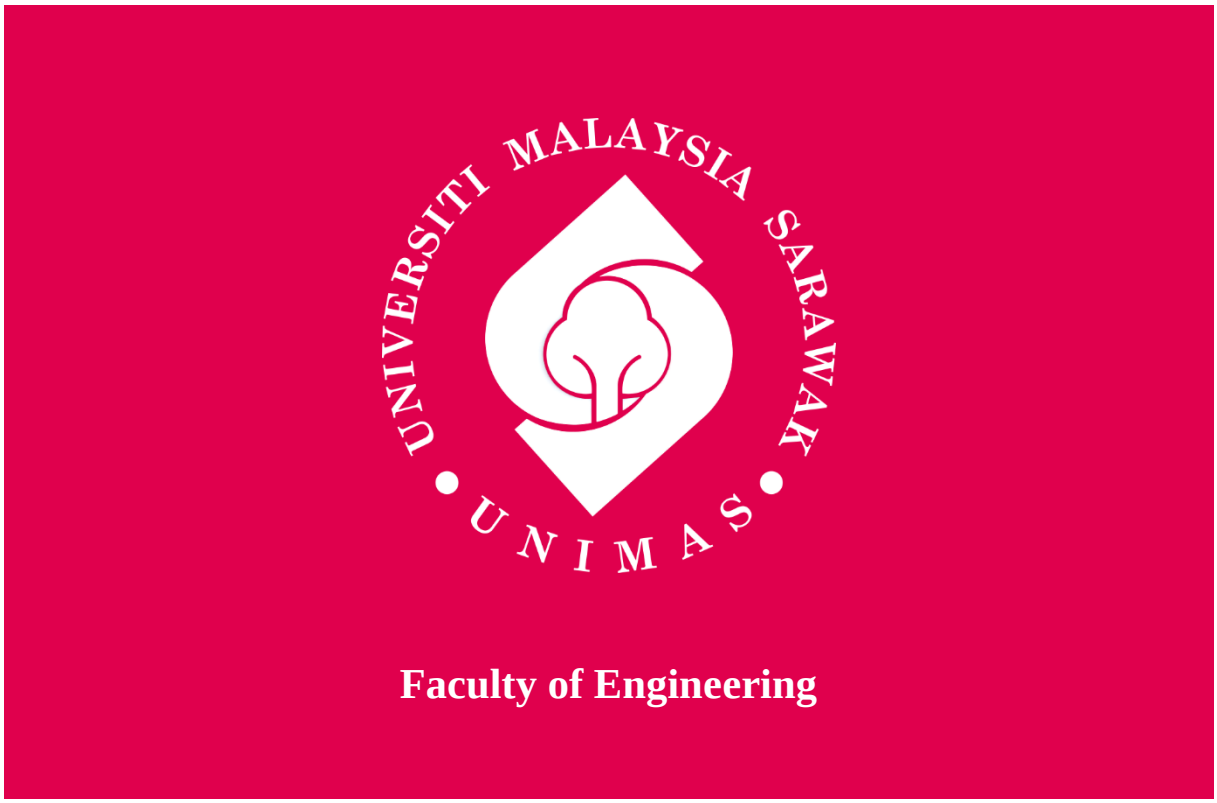

Modelling of Data-Driven Monotone Zero-Order TSK Fuzzy Inference Systems

Teh Chin Ying

Doctor of Philosophy
2018

Modelling of Data-Driven Monotone Zero-Order TSK Fuzzy Inference System

Teh Chin Ying

A thesis submitted

In fulfilment of the requirements for the degree of Doctor of Philosophy

(Electronic Engineering)

Faculty of Engineering
UNIVERSITI MALAYSIA SARAWAK
2018

DECLARATION

I, Teh Chin Ying (13010063) from Faculty of Engineering UNIMAS hereby declare that the work entitled, Modelling of Data-Driven Monotone Zero-Order TSK Fuzzy Inference Systems is my original work. I have not copied from any other students' work or from any other sources except where due reference or acknowledgement is made explicitly in the text, nor has any part been written for me by another person. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

_____

Teh Chin Ying (13010063)

Date: 24-09-2018

ACKNOWLEDGEMENT

First and foremost, I would like to express my brilliant thanks to my supervisor, Dr. Tay Kai Meng for his guidance, inspiration, support and patience throughout this study. His advice helped me think rationally about complex problems and solve them in a correct and yet simple manner. His guidance has nurtured the seed of research in me and has motivated me a lot in completing this thesis. His contribution is noted with valued.

Special thanks to the Kementerian Pendidikan Tinggi (KPT), Malaysia for the MyBrain15 scholarship (MyPhD), referrer code: KPT(B) 880410075194 for granting me the scholarship which covers my tuition fees and supports my living expenses during the term of this study.

My sincere thanks to all my friends and colleagues who never fail to help and encouraged me during my study. Last but not least, I wish my appreciation and thanks to my beloved family for their endless patience, love, motivation, understanding and encouragement along the way aspire me to venture and finally completed my research. Thank you.

ABSTRACT

In this thesis, the use of two data-driven Fuzzy Inference System (FIS) model, i.e., ad hoc and system identification methods in the literature are analyzed. The data-driven FIS model is used to develop monotone fuzzy If-Then rules based on single and multi-attribute data for modelling a zero-order Takagi-Sugeno-Kang (TSK) FIS model. Convex and normal trapezoidal (triangular) fuzzy sets coupled with a strong fuzzy partition strategy are employed. Firstly, a monotone TSK FIS model that can always be obtained using a set of complete and monotone fuzzy If-Then rules is explained. A monotone test is devised to evaluate the degree of monotonicity (DOM) of TSK FIS models. The analysis further indicates that with single-attribute monotone data, monotone fuzzy If-Then rules can be produced using both ad hoc and system identification methods. However, this is not the case for multi-attribute monotone data. This finding is important for developing a monotone TSK FIS model, as it highlights that even with a "clean" monotone data set, the generated fuzzy If-Then rules may need to be pre-processed (i.e., re-labeled), in order to guarantee the monotonicity property of the resulting TSK FIS model. As such, a framework for designing and developing monotonicity-preserving FIS models for ad hoc and system identification methods are proposed, respectively. The effectiveness of the proposed approach is evaluated using a number of simulated examples and a real-world benchmark problem. The results indicate that the proposed approach coupled with a monotone data set leads to a lower mean square error and a better DOM score.

Keywords: Degree of monotonicity; monotone data; monotone fuzzy rules; monotone fuzzy rules relabeling; monotonicity test; strong fuzzy partition; system identification; TSK fuzzy inference system

Pemodelan Sistem yang Dipandu oleh Data secara Monoton untuk Sistem Inferensi Kabur Tertib Sifar TSK

ABSTRAK

Dalam tesis ini, penggunaan dua Sistem Inferensi Kabur yang didorong oleh data, iaitu kaedah ad hoc dan pengenalan sistem adalah dianalisis. Penggunaan Sistem Inferensi Kabur yang didorong oleh data adalah untuk membangunkan kaedah-kaedah monoton Jika-Kemudian kabur berdasarkan data tunggal dan multi-attribut untuk pemodelan model Takagi-Sugeno-Kang (TSK) FIS. Convex dan normal trapezoid (segi tiga) kabur set serta menggunakan strategi pemisahan kabur yang kuat. Pertama, model monoton TSK FIS yang boleh diperolehi dengan menggunakan satu set lengkap dan peraturan monoton kabur Jika-Kemudian dijelaskan. Ujian monoton dibuat untuk menilai tahap monotonisme (DOM) model TSK FIS. Analisis itu selanjutnya menunjukkan bahawa dengan data monoton tunggal, peraturan-peraturan monoton Jika-Kemudian kabur boleh dihasilkan menggunakan kaedah ad hoc dan pengenalan sistem. Walau bagaimanapun, ini tidak berlaku untuk data monoton berbilang sifat. Dapatan ini adalah penting untuk membangun model monoton TSK FIS, kerana ia menyoroti bahawa walaupun dengan set data monoton yang "bersih", peraturan-peraturan Jika-Kemudian kabur yang dijana mungkin perlu diproses terlebih dahulu (iaitu, dilabel semula), untuk menjamin sifat monotonisme model TSK FIS yang terhasil. Oleh itu, satu rangka kerja untuk merekabentuk dan membangunkan monotonisme yang memelihara model FIS untuk kaedah ad hoc dan pengenalan sistem dicadangkan masing-masing. Keberkesanan pendekatan yang dicadangkan dinilai menggunakan beberapa contoh simulasi dan masalah penanda aras dunia sebenar. Hasilnya menunjukkan bahawa pendekatan yang dicadangkan ditambah

dengan set data monoton membawa kepada kesilapan rata min yang lebih rendah dan skor DOM yang lebih baik.

***Kata Kunci**: Ijazah monotonik; data monoton; peraturan kabur monoton; peraturan kabur monoton pelabelan; ujian monotetik; partisi kabur yang kuat; pengenalan sistem; TSK sistem inferensi kabur*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AARS	Approximate Analogical Reasoning Scheme
ALARP	As Low As Reasonably Practicable
ANFIS	Adaptive Neuro Fuzzy Inference System
AR	Analogical Reasoning
Auto-MPG	Auto-Mile Per Gallon
DFP	Davidon-Fletcher-Powell
DOM	Degree of Monotonicity
FATI	First Aggregate Then Inference
FCMC	Fuzzy c-Means Clustering
FERI	Fundamental Equation of Rule Interpolation
FIS	Fuzzy Inference System
FITA	First Inference Then Aggregate
FMEA	Failure Mode Effect Analysis
FPR	Fuzzy Production Rule
Fre	Frequency
FRI	Fuzzy Rule Interpolation
FST	Fuzzy Set Theory
HS	Huang and Shen
KH	Kóczy and Hirota
LSE	Least Square Estimator
MACI	Modified Alpha Cut Interpolation
MF	Membership Function

MFRR	Monotone Fuzzy Rule Relabeling
MOI	Mean of Inversion
MSE	Mean Square Error
NN	Neuro Networks
PMV	Predicted Mean Vote
RPN	Risk Priority Number
Sev	Severity
SIRM	Single Input Rule Model
TSK	Takagi-Sugeno-Kang
WFPR	Weighted Fuzzy Production Rule
WM	Wang-Mendel

CHAPTER 1

INTRODUCTION

1.1 Research Background

Inference is a process of drawing a conclusion by applying heuristics (based on logic, statistics, etc.) to observations or hypotheses; or by interpolating the next logical step in an intuited pattern (Kneebone, 2001; Russell & Norvig, 2003). There are two main types of inference, i.e., deductive inference and inductive inference. On one hand, in deductive inference, the truth of the premises guarantees the truth of the conclusion. It is impossible for the premises to be true and yet the conclusions to be false (Kahane, 1990). On the other hand, inductive inference is the process of reaching a general conclusion from specific examples (Kahane, 1990; Russell & Norvig, 2003).

An inference technique is a method that attempts to derive answers from a knowledge base. It can be viewed as the "brain" that reasons about the information in the knowledge base for the ultimate purpose of formulating new conclusions (Russell & Norvig, 2003). From the literature, a number of inference techniques have been studied, e.g. Bayesian inference (Box & Tiao, 2011; Dempster, 1968), optimal perceptual inference (Hinton & Sejnowski, 1983), automatic logical inference (Harrison, 2009), probabilistic inference (Cooper, 1990; Dagum & Luby, 1993; Pearl, 1988), and fuzzy inference (Jang, Sun, & Mizutani, 1997).

The focus of this thesis is on the Fuzzy Inference System (FIS), which have been successfully applied in a wide variety fields. Examples of successful application of FIS models include modelling (Du & Zhang, 2008; Jang et al., 1997; Lin & Lee, 1995), classification (Jang et al., 1997; Lin & Lee, 1995; Sengur, 2008), decision analysis

(Oderanti & De Wilde, 2010), and control (Feng, 2006; Kurnaz, Cetin, & Kaynak, 2010) problems. An FIS model can be viewed as a computing paradigm based on the concepts of fuzzy set theory, fuzzy production rule (If-Then rule), and fuzzy reasoning (Jang et al., 1997). Owing to its multidisciplinary nature, FIS is also knows by numerous others name, such as fuzzy-rule-based system, fuzzy expert system (Kandel, 1992), fuzzy model (Sugeno & Kang, 1988; T. Takagi & Sugeno, 1985), fuzzy associative memory (Kosko, 1990), fuzzy logic controller (Lee, 1990a, 1990b; Mamdani & Assilian, 1975) and simply (and ambiguously) fuzzy system. Examples of popular FIS models include the Mamdani FIS (Mamdani & Assilian, 1975), Sugeno/TSK FIS (Sugeno & Kang, 1988; T. Takagi & Sugeno, 1985), and Tsukamoto FIS (Tsukamoto, 1979). The differences between these three fuzzy inference systems depends on the consequents of their fuzzy rules, therefore, their aggregation and deffuzification procedure are differ respectively.

The success of FIS models is largely owing to the following key factors: (i) they are able to utilize linguistic information from human experts (Jang et al., 1997; Lin & Lee, 1995; Wang, 1992); (ii) they are able to simulate human thinking (Zadeh, 1973); (iii) they are able to capture approximate and inexact nature (i.e., uncertainty) of the real world (Jang et al., 1997; Lin & Lee, 1995; Wang, 1992); (iv) they can be expressed with linguistic variables, which can easily be interpreted by humans (Jang et al., 1997; Lin & Lee, 1995); (v) they are able to act as universal approximator to approximate any real continuous functions to any degree of accuracy (Kosko, 1994; Wang, 1992).

In some cases, the fuzzy If-Then rules are obtained from human experts, i.e., knowledge-driven FIS models. Owing to the difficulty to obtain fuzzy rules from human experts, many methods automatically generate fuzzy If-Then rules from numerical data,

i.e., data-driven FIS models, have been proposed. Among the popular data-driven methods include system identification (Jang et al., 1997), gradient descent learning (Ichihashi & Watanabe, 1990; H. Ishibuchi, Nozaki, Tanaka, Hosaka, & Matsuda, 1993; Jang, 1993; Nomura, Hayashi, & Wakami, 1992), fuzzy-neural (H. Takagi & Hayashi, 1991), which are often coupled with fuzzy *c*-means clustering (Sugeno & Yasukawa, 1993), least-square (Sugeno & Kang, 1988; T. Takagi & Sugeno, 1985), and *ad hoc* methods (Hisao Ishibuchi, Nozaki, Tanaka, Hosaka, & Matsuda, 1994; Nozaki, Ishibuchi, & Tanaka, 1997; Wang, 2003; Wang & Mendel, 1992).

The focus of this thesis is on the data-driven FIS models using (i) *ad hoc* methods and (ii) system identification method. For *ad hoc* methods, there are two different methods to produce data-driven FIS models (Nozaki et al., 1997), i.e., (i) Wang et al. (Wang, 2003; Wang & Mendel, 1992) (hereafter abbreviated as the Wang-Mendel (WM) method); (ii) Ishibuchi et al. (Hisao Ishibuchi et al., 1994) (hereafter abbreviated as the Ishibuchi method). The classification terminologies in (Jorge Casillas, Cordón, & Herrera, 2002), in which both WM (Wang, 2003; Wang & Mendel, 1992) and Ishibuchi (Hisao Ishibuchi et al., 1994; Nozaki et al., 1997) methods are categorized as "*ad hoc* methods".

1.2 Problem Statements and Motivations

Even though data-driven FIS models are popular, research on *data-driven monotone FIS models* is new. In this regard, the importance of the monotonicity property in FIS modeling has been highlighted in a number of recent publications. For example, a number of methods to preserve the monotonicity property and to realize their applications have also been proposed. These include the development of mathematical conditions as a set of governing equations (Broekhoven & Baets, 2009; Seki, Ishii, & Mizumoto, 2010;

Won, Park, & Lee, 2002), real-world problem-solving using the developed mathematical conditions (Tay & Lim, 2011c), as well as extension and synthesis of the mathematical conditions with advanced FIS modeling techniques (Husek, 2016; Kouikoglou & Phillis, 2009; Tay & Lim, 2011b).

Besides that, to the best of our knowledge, investigations on modeling of monotonicity property pertaining to FIS models based on monotone data set (clean data samples that are free from noise) are new. It is not clear, if monotone data set can always generate monotone fuzzy rules, using both *ad hoc* and system identification methods. Therefore, the hypothesis of this thesis is to ascertain whether both *ad hoc* and system identification methods are able to always produce monotone fuzzy If-Then rules with single and multi-attribute monotone data sets. This hypothesis is very important and useful in practical applications, because given a "clean" data set, the generated fuzzy If-Then rules from *ad hoc* and system identification methods may not always satisfy the monotonicity property pertaining to FIS modelling.

Thus, an analytical study is conducted to indicate where the fuzzy If-Then rules generated using *ad hoc* and system identification methods are able to always produce monotone fuzzy If-Then rules with single and multi-attribute monotone data sets. In addition, an approach to design and develop a monotonicity property pertaining to FIS modelling with both *ad hoc* and system identification methods is examined and investigated in details.

1.3 Research Methodology

The methodology adopted in this research is depicted in Figure 1.1. First, the background and literature review on related theory and operations of FIS models are

described. The *sufficient condition* for the monotone FIS model is established. A monotone test is established to evaluate the monotonicity property of an FIS model. An analysis for the data-driven FIS models, i.e., *ad hoc* and system identification methods are conducted. A monotonicity data-driven preserving FIS modelling approach for both *ad hoc* and system identification methods are developed. The proposed approach is applied to several practical FIS modelling problems, i.e., (i) an FIS-based control problem; (ii) an FIS-based auto mile per gallon (Auto-MPG) problem and (iii) an FIS-based ecology problem. A monotonicity-preserving FIS-based control problem is first proposed and examined. An empirical experiments with data generated from the Predicted Mean Vote (PMV) index are presented. Then, a monotonicity-preserving FIS-based Auto-MPG problem is proposed and examined. The use of the proposed procedure in FIS-based ecology problems is also investigated. Finally, conclusions from this research are drawn, and suggestions for further work are presented.

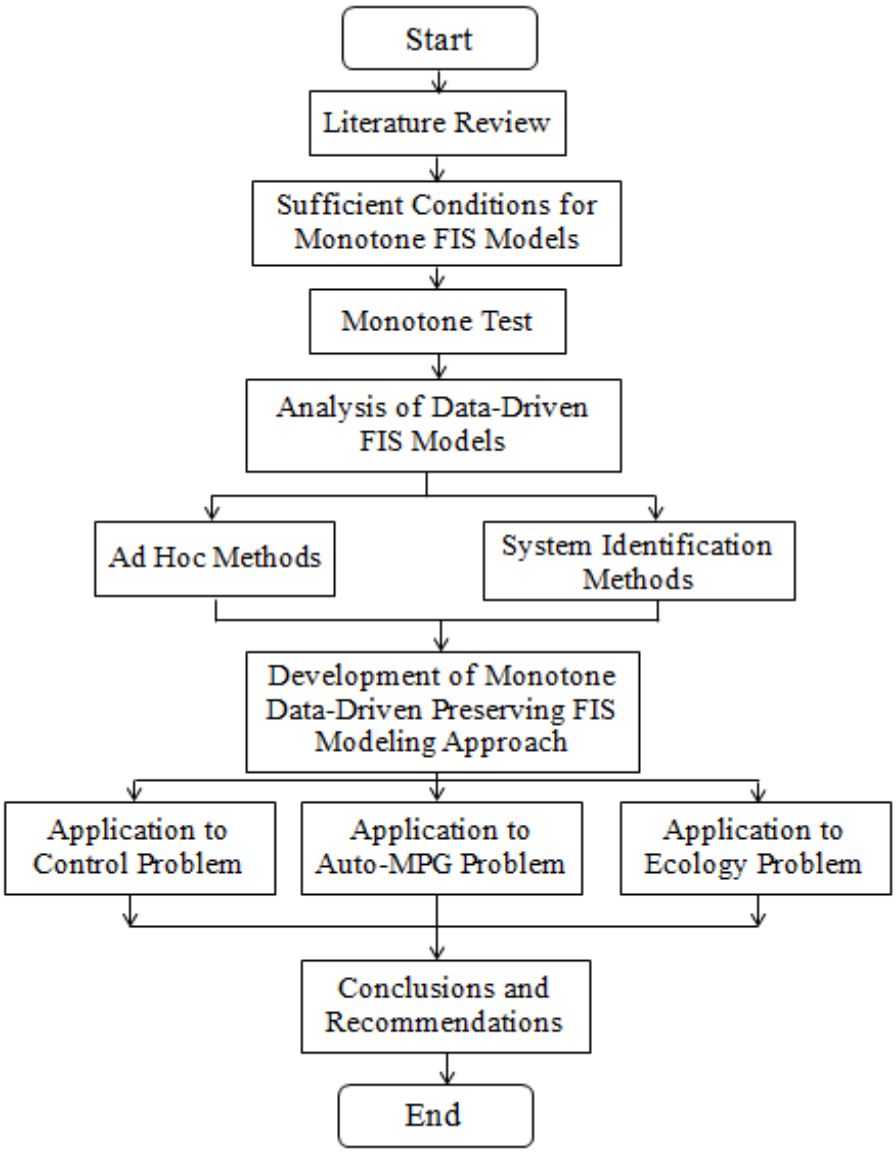

Figure 1.1: Research methodology

1.4 Objectives of the Research

The main aim of this research is to investigate whether both *ad hoc* and system identification methods are able to always produce monotone fuzzy If-Then rules with single and multi-attribute monotone data sets. A set of *sufficient conditions* and a monotone test are established to evaluate the monotonicity property of an FIS model. A novel framework for designing and developing monotonicity-preserving FIS models for *ad*

hoc and system identification methods is proposed and the applicability of the proposed approach in resulting monotonicity-preserving FIS models is examined. The specific objectives are as follows.

- To investigate the use of the *sufficient conditions* as a set of governing equation for designing and developing monotonicity-preserving FIS models.
- To analysis whether both *ad hoc* and system identification methods are able to always produce monotone fuzzy If-Then rules with single and multi-attribute monotone data sets
- To design and model monotonicity-preserving FIS models for *ad hoc* and system identification methods.
- To demonstrate the applicability of the resulting FIS models to various problems in the domains of control, Auto-MPG and ecology.

1.5 Scope of the Research

In this thesis, the *sufficient condition* is proposed as a set of governing equation for designing and developing monotonicity-preserving FIS models. A monotone test is established to evaluate the monotonicity property of an FIS model. The effectiveness of the *sufficient condition* for modelling an FIS model that preserves the monotonicity property is demonstrated using two FIS-based model, i.e., FIS-based risk model and FIS-based occurrence model. An analysis in conducted to ascertain whether *ad hoc* and system identification methods are able to always produce monotone fuzzy If-Then rules with single and multi-attribute monotone data sets. As the analytical results indicate that fuzzy If-Then rules generated using *ad hoc* methods and the empirical results indicate that fuzzy If-Then rules generated using system identification method do not always satisfy the

monotonicity property when a multi-attribute monotone data set is considered. A framework for designing and developing monotonicity-preserving FIS models for *ad hoc* and system identification methods are proposed, respectively. The effectiveness of the proposed approach is demonstrated using three FIS-based applications, i.e., control, Auto-MPG and ecology problem.

1.6 Organization of the Thesis

This thesis is organized as follows. In this introductory chapter, the research background is first described. The problem statement and motivations are explained. The research methodology, objectives, and scope are also presented.

In Chapter 2, the background and literature review on fuzzy sets theory, fuzzy sets theoretical operations, fuzzy production rule (FPR), fuzzy reasoning, fuzzy inference system (FIS) models, and fuzzy rule interpolation (FRI) techniques are presented. The background and literature review on data-driven FIS models are presented too. The literature review covers mainly the monotonicity property of the FIS models. In Chapter 3, a *sufficient condition* for the zero-order TSK FIS model is established. A monotone test is established to evaluate the monotonicity property of an FIS model. The effectiveness of the *sufficient condition* for modelling an FIS model that preserves the monotonicity property is demonstrated using two FIS-based models, i.e., FIS-based risk model and FIS-based occurrence model, respectively.

In Chapter 4, an analysis for two *ad hoc* methods are conducted to ascertain whether both *ad hoc* methods are able to always produce monotone fuzzy rules with single and multi-attribute *monotone* data sets. The analytical results indicate that fuzzy If-Then rules generated using *ad hoc* methods do not always satisfy the monotonicity property

when a multi-attribute monotone data set is considered. Therefore, a framework for designing and developing monotonicity-preserving FIS models for *ad hoc* methods are proposed, respectively. A number of simulated examples are used to demonstrate the usefulness of the proposed framework for modeling monotone TSK FIS models with *ad hoc* methods. In Chapter 5, an application of the proposed framework for *ad hoc* methods to prediction of the thermal comfort index is reported. In addition, two real-world benchmark problems, i.e., Auto MPG problem from the UCI Machine Learning Repository and the mercury-in-fish problem from International Council for the Exploration of the Sea, are used for experimentations too.

In Chapter 6, the use of a system identification-based framework to develop monotone fuzzy If-Then rules using multi-attribute data for modelling a zero-order TSK FIS model is analysed. The analysis for a system identification method is conducted to ascertain whether the system identification method could always produce monotone fuzzy rules with multi-attribute monotone data sets. The analytical results indicate that fuzzy If-Then rules generated using system identification methods do not always satisfy the monotonicity property when a multi-attribute monotone data set is considered. Therefore, a framework for designing and developing monotonicity-preserving FIS models for system identification methods are proposed, respectively. A series of benchmark problems used in Chapter 5 are re-used for evaluation to demonstrate the usefulness of our proposed system identification-based framework for developing monotone TSK FIS models with system identification method.

Finally, concluding remarks and contributions of this research are presented in Chapter 7. Suggestions for further works are also included.

CHAPTER 2

LITERATURE REVIEWS

2.1 Chapter Overview

In the previous chapter, an introduction to the research is presented. In this chapter, the background and literature review on fuzzy sets theory, fuzzy sets theoretical operations, Fuzzy Production Rule (FPR), fuzzy reasoning, Fuzzy Inference Systems (FISs), and Fuzzy Rule Interpolation (FRI) techniques are presented. The background and literature review on data-driven FIS models are presented too. A review on the monotonicity property of FIS is further described. Note that the literature review is mainly focused on theoretical aspect of FIS models. Other related literature reviews, especially those on the application of FIS models, are presented in the appropriate sections in subsequent chapters.

In Section 2.2, the fuzzy sets theory and fuzzy sets theoretical operations are reviewed. In Section 2.3, the fuzzy rule base models, i.e., linguistic variables, Fuzzy Production Rule (FPR), fuzzy reasoning, and Fuzzy Inference Systems (FISs) are reviewed. In Section 2.4, the background and the literature review on data-driven FIS models are presented. In Section 2.5, the background and the literature review on Fuzzy Rule Interpolation (FRI) are presented. In Section 2.6, the monotonicity property of FIS models and the related researches are discussed. Finally, concluding remarks are presented in Section 2.8.

2.2 Fuzzy Sets

2.2.1 Fuzzy Sets Theory (FST)

The theory of sets as a mathematical discipline was introduced by a German mathematician, G. Cantor (1845-1918) (Stoll, 1975). Cantor suggested that a set is made up of objects called members or elements, and one can determine whether or not an object is a member of a set (Stoll, 1975). Let X be a space of objects and x be a generic element of X. A classical (crisp) set is normally defined as a collection of elements or objects $x \in X$. Each element can either belong to or not belong to a set A, $A \subseteq X$. In the former case, the statement "x belongs to A" is true. On the other hand, in the latter case, the statement "x belongs to A" is false. The characteristic function or membership function of a set for each element x in X can be represented with Equation 2.1.

$$\mu_A(x) = \begin{cases} 1 & if\ and\ only\ if\ x \in A \\ 0 & if\ and\ only\ if\ x \notin A \end{cases} \tag{2.1}$$

A fuzzy set, on the other hand, is an extension of the concept of a crisp set which introduces vagueness by eliminating the sharp boundary that divides members from non-members in the group (Zadeh, 1965). A fuzzy set A in X is characterized by a characteristic function or membership function, $\mu_A(x)$ which is allowed to have a real number in the interval between 0 and 1, with the value of $\mu_A(x)$ at x representing the degree of membership of x in A. For example, a fuzzy set, A, in X is defined as a set of ordered pairs: $A = \{(x, \mu_A(x)) | x \in X\}$ where $\mu_A(x)$ is the membership function of x in A. The nearer the value of $\mu_A(x)$ to unity, the higher the degree of membership of x in A.

Several types of membership functions can be used to represent a fuzzy set, such as triangular, trapezoidal and Gaussian membership function (Jang et al., 1997). The triangular membership functions of fuzzy set A, $\mu_A(x)$ is a continuous increasing functions from a to b, then $\mu_A(x)$ is a continuous decreasing function from b to c. The continuous function in x is presented in Equation 2.2.

$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \le x \le b, \\ \dfrac{c-x}{c-b}, & b \le x \le c, \\ 0. & otherwise. \end{cases} \tag{2.2}$$

where, a, b, and c are any real number that represent the x coordinates of the three vertices of $\mu_A(x)$ in a fuzzy set A. In Figure 2.1, the triangular membership function of fuzzy set A, $\mu_A(x)$ is presented.

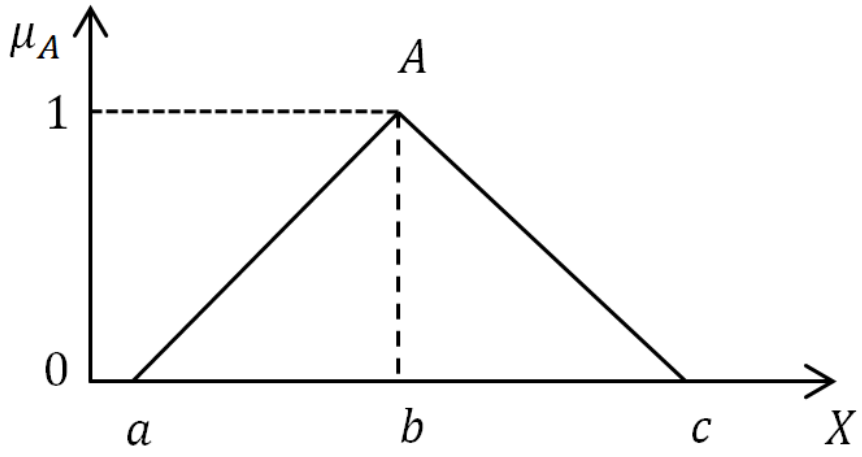

Figure 2.1: Triangular membership function

The trapezoid membership function of fuzzy set A, $\mu_A(x)$ is a continuous increasing function from a to b, $\mu_A(x)$ is a constant value where $\mu_A(x) = 1$ from b to c

and $\mu_A(x)$ is a continuous decreasing function from c to d. Trapezoidal curves depend on four parameters and the continuous function in x is presented in Equation 2.3.

$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \le x \le b, \\ 1, & b \le x \le c, \\ \dfrac{d-x}{d-c}, & c \le x \le d, \\ 0. & otherwise. \end{cases} \tag{2.3}$$

where, a, b, and c, d are any real number. In Figure 2.2, the trapezoidal membership function of fuzzy set A, $\mu_A(x)$ is presented.

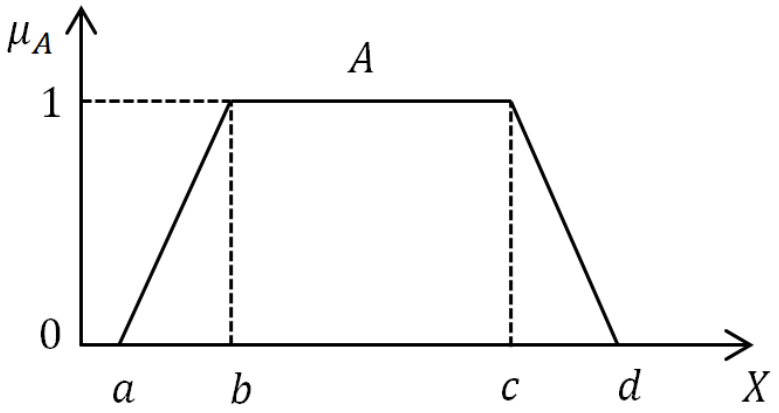

Figure 2.2: Trapezoidal membership function

2.2.2 Fuzzy Set Theoretic Operations

Union, intersection and complement are the most basic operations on classical sets. Corresponding to these three operations, fuzzy sets have similar operations, as defined by Zadeh (1965).

The union of two fuzzy sets, A and B, is a fuzzy set C, written as $C = A \cup B$ or $C = A \text{ OR } B$. The membership function C of the union A and B is pointwise defined

by $\mu_C(x) = \mu_A(x) \vee \mu_B(x)$. Jang *et al.* (1997) listed several frequently used union operators, as follows.

Maximum: $$\mu_C(x) = \max\left(\mu_A(x), \mu_B(x)\right) = \mu_A(x) \vee \mu_B(x)$$

Algebraic product: $$\mu_C(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$$

Drastic product: $$\mu_C(x) = \begin{cases} \mu_A(x) & if\ \mu_B(x) = 0 \\ \mu_B(x) & if\ \mu_A(x) = 0 \\ 0 & if\ \mu_A(x), \mu_B(x) < 0 \end{cases}$$

The intersection of two fuzzy sets, A and B, is a fuzzy set C, written as $C = A \cap B$ or $C = A$ AND B. The membership function C of the intersection A and B is pointwise defined by $\mu_C(x) = \mu_A(x) \wedge \mu_B(x)$. Again, Jang *et al.* (1997) listed several frequently used intersection operators, as follows.

Minimum: $$\mu_C(x) = \min\left(\mu_A(x), \mu_B(x)\right) = \mu_A(x) \wedge \mu_B(x)$$

Algebraic product: $$\mu_C(x) = \mu_A(x) \times \mu_B(x)$$

Drastic product: $$\mu_C(x) = \begin{cases} \mu_A(x) & if\ \mu_B(x) = 1 \\ \mu_B(x) & if\ \mu_A(x) = 1 \\ 0 & if\ \mu_A(x), \mu_B(x) < 1 \end{cases}$$

The complement of fuzzy set A is denoted by $\bar{A}$ ($NOT\ A$). The membership function of $\bar{A}$, can be written as $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

2.3 Fuzzy Rule Base Model

2.3.1 Linguistic Variables

A linguistic variable is characterized by a quintuple $(x, T(x), X, G, S)$ in which x is the name of the variable; $T(x)$ is the term set of x, that is, the set of names of linguistic values of x; X is the universe of discourse; G is a syntactic rule which usually has the form of a grammar for generating the name, X, of values of x; and S is a semantic rule which associates with each linguistic value A its meaning $S(A)$, where $S(A)$ denotes a fuzzy set in X.

For an example, If *"Age"* is interpreted as a linguistic variable, then its term set $T(Age)$ could be $T(Age) = \{very\ young, young, old, very\ old\}$ where each term in $T(Age)$ is characterized by a fuzzy set of universe of discourse $X = [0,100]$. Usually, *"Age is young"* is denoted as *"young"* for the linguistic variable of age. Alternatively, when age is interpreted as a numerical variable, the expression *"Age=30"* is assigned instead of defining the numerical value *"30"* to the numerical variable age. The syntactic rule refers to the way the linguistic values in the term set $T(Age)$ are generated. The semantic rule defines the membership function of each linguistic value of the term set.

2.3.2 Fuzzy Production Rules (Fuzzy IF-THEN Rules)

A major component of an FIS model is the fuzzy production rules (FPR) (Jang et al., 1997; Mendel, 1995). An FPR is expressed as a logical implication, i.e., in a form of an If-Then statement. A general fuzzy If-Then rule has the form, "If x is A then y is B" where A and B are the linguistic values defined by the fuzzy sets on universe of discourse X and Y, respectively. "x is A" is known as antecedent or premise, while "y is B" is

known as consequence or conclusion. It is a form of proposition, whereby a proposition is an ordinary statement involving terms which has been defined, e.g. "the damping ratio is low" (Mendel, 1995). From the proposition, the relevant rule can be obtained: "IF the damping ratio is low THEN the system's impulse response oscillates a long time before it dies out". Propositions can be combined or modified in many ways, via the set-theoretic operations, i.e., AND, OR, and NOT. There are several strategies to partition the input space to form the antecedent. Among them are grid partition, tree partition, and scatter partition (Jang et al., 1997; Lin & Lee, 1995).

Among the partition strategies, the grid partition is well known and often chosen for designing FIS models (Jang et al., 1997). With the grid partition, an FPR with *n* antecedents has the form: $If\ (x_1 is\ A_1) AND\ (x_2 is\ A_2)\ \dots AND (x_n is\ A_n), THEN (y\ is\ B)$, where x_i and y are the inputs and output of the FIS model, A_1, A_2,…A_n and B are linguistic variables/fuzzy sets for the inputs and output, respectively.

A Weighted Fuzzy Production Rule (WFPR) is an enhancement of an FPR. A WFPR allows knowledge imprecision to be taken into account by adding extra knowledge representation parameters, which include threshold value, certainty factor, local weight, and global weight (Lau & Chan, 1997; Tsang & Yeung, 1997). Generally, a WFPR with *n* antecedents can be represented by:

$$If\ (x_1 is\ A_1) AND\ (x_2 is\ A_2)\ \dots AND (x_n is\ A_n), THEN (y\ is\ B)$$

$$CF_R, LW_1, LW_2, \dots, LW_n, \lambda_{A_1}, \lambda_{A_2}, \dots, \lambda_{A_n}, GW$$

$$Fact_i: x_i\ is\ A_i'\ with\ CF_i, 1 \le i \le n$$

$$Conclusion: Y\ is\ B'\ with\ CF_{B'}$$

The parameters are explained as follows. A threshold value, $\lambda_{A_i}$is assigned to a proposition. It ensures that the degree of similarity between the proposition ($x_i\ is\ A_i$) and its fact, i.e., greater than or equal to λ_{A_i}. The assignment of λ_{A_i} to "$x_i\ is\ A_i$ " is not only to ensure the result of an approximate reasoning method is reasonable but also to prevent or reduce rule mis-firing (Tsang & Yeung, 1997). The certainty factor for a given fact ($Fact_i$), CF_i determines how certain the proposition is. It is used to express how accurate, truthful, or reliable the fact is. The certainty factor can also be applied to a rule (CF_R). It means how certain the relationship the antecedent and the consequent is (Tsang & Yeung, 1997).

For an FPR that comprises more than a proposition connected by "AND", the local weight for a proposition ($x_i\ is\ A_i$), LW_i is used to indicate the degree of importance of the proposition in relation to the antecedent (Tsang & Yeung, 1997). Global weight, GW, is used to indicate the degree of importance of each rule's contribution to the final goal. There are two different applications of the global weight (Tsang & Yeung, 1997): (i) to compare the relative degree of importance of a particular rule with those from other rules in a given inference path leading to a specific output membership function; (ii) to show the relative importance of a rule when it is used in different inference paths leading to different output membership functions.

2.3.3 Fuzzy Reasoning

In 1975, Zadeh introduced the theory of approximate reasoning (also known as fuzzy reasoning) (Zadeh, 1975). This idea provides a powerful framework for reasoning that imprecise and uncertain information. Fuzzy reasoning is an inference procedure that derives conclusions from a set of fuzzy If-Then rules. It can be written as

Fact:	x is A',
FPR:	If x is A then y is B,
Consequence:	y is B'

where A' is close to A, and B' is close to B.

2.3.4 The Zero-Order TSK Fuzzy Inference System (FIS)

An FIS model can be explained as a computing paradigm that is based on the concepts of fuzzy set theory, FPRs, and fuzzy reasoning. Consider an FIS model with n inputs. Let $\boldsymbol{x} = \left(x_1, x_{2,} \cdots, x_n\right)$ be the inputs vector. Consider k_i terms at the i^{th} input space, $A_{i,1}, A_{i,2}, \cdots, A_{i,k_i}$ which are represented by fuzzy membership functions $\mu_{i,1}(x_i), \mu_{i,2}(x_i), \cdots, \mu_{i,k_i}(x_i)$, respectively. The output of the FIS model, $y = f(\boldsymbol{x})$. If a full grid partition is used, the number of fuzzy rule is $\prod_{i=1}^{n} k_i$.

The FPR of a single-attribute $(n = 1)$ FIS model, i.e., $R_j: A_j \rightarrow b_j$ where $j = 1,2 \cdots, k$ is presented in Equation 2.4.

$$\begin{array}{c} R_1\text{: IF } x \text{ is } A_1 \text{ THEN } y \text{ is } b_1 \\ R_2\text{: IF } x \text{ is } A_2 \text{ THEN } y \text{ is } b_2 \\ \vdots \\ R_k\text{: IF } x \text{ is } A_k \text{ THEN } y \text{ is } b_k \end{array} \tag{2.4}$$

Note that A_1 , A_2 , $\cdots$, and A_k are linguistic terms at the rule antecedent part, and are represented by membership function $\mu_1(x), \mu_2(x), \cdots,$ and $\mu_k(x)$, where $j = 1,2, \cdots, k$ respectively; $b_1, b_2, \ldots, b_k$ are fuzzy singletons at the rule consequent part, respectively. The output of the FIS model is obtained using an inference technique is presented in Equation 2.5.

$$y = f(x) = \frac{\sum_{j=1}^{k} \mu_j(x) \times b_j}{\sum_{j=1}^{k} \mu_j(x)} \tag{2.5}$$

The FPR of a multi-attribute $(n > 1)$ FIS model, i.e., $R_{\boldsymbol{j}} : A_{\boldsymbol{j}} \rightarrow b_{\boldsymbol{j}}$ where $\boldsymbol{j} = j_1, j_2, \cdots, j_n$ and $j_i = 1,2, \cdots, k_i$ is presented as follows is presented in Equation 2.6 (Liang & Mendel, 2000).

If x_1 is A_{1,j_1} AND x_2 is A_{2,j_2} ... AND x_n is A_{n,j_n}, THEN y is $b_{j_1,j_2,\cdots,j_n} = b_{\boldsymbol{j}}$ (2.6)

Consider the AND operator as the product function. The output is obtained by using the weighted average of a representative real value, $b_{\boldsymbol{j}}$, with respect to its compatibility grade. The respective equation is presented in Equation 2.7.

$$y = f(\boldsymbol{x}) = \frac{\sum_{j_n=1}^{k_n} \cdots \sum_{j_1=1}^{k_1} \left[\mu_{1,j_1}(x_1) \times \cdots \times \mu_{n,j_n}(x_n)\right] \times b_{\boldsymbol{j}}}{\sum_{j_n=1}^{k_n} \cdots \sum_{j_1=1}^{k_1} \left[\mu_{1,j_1}(x_1) \times \cdots \times \mu_{n,j_n}(x_n)\right]} \tag{2.7}$$

FIS models can be classified into two categories: First Inference Then Aggregate (FITA) and First Aggregate Then Inference (FATI) (Cordón, Herrera, & Peregrín, 1997; Emami, Turksen, & Goldenberg, 1999; H. Ishibuchi, Yamamoto, & Nakashima, 2006). Equations 2.6 and 2.7 belong to FITA (H. Ishibuchi et al., 2006). For an FITA model, the representative value is first determined. Then, the output estimate is obtained by aggregating the crisp values of the compatibility fuzzy rules. The weighted average is one of the methods to obtain the output estimate. An FIS model that uses a fuzzy set or a crisp value at its rule consequent as in Equations 2.6 and 2.7 is categorized as an FIS model with a high degree of interpretability (Casillas, Cordón, Herrera, & Magdalena, 2003) which

allows direct translation of the rules. Besides, Equations 2.6 and 2.7 represent a singleton, zero-order Sugeno FIS model. The associated fuzzy reasoning method has several advantages, e.g., its reasoning mechanism is simple and it is suitable for gradient-based learning algorithm (Ishibuchi et al., 2006)

2.3.5 Recent Advances on FIS Modelling

Over the years, researches to enhance FIS models have been reported. Examples include fuzzy systems with neural network learning, e.g. Adaptive Neuro Fuzzy Inference System (ANFIS) (Jang, 1993; Jang & Sun, 1995), and with evolutionary computation learning (Ishibuchi, Nozaki, Yamamoto, & Tanaka, 1995). A type-two FIS model that incorporates type-two fuzzy sets (Zadeh, 1975) has been investigated in (Karnik, Mendel, & Liang, 1999; Liang & Mendel, 2000). An FIS model with a rule reduction technique based on a similarity measure and with interpretability improvement has been suggested in (Jin, 2000). Other advances include the development of Analogical Reasoning (AR) techniques (Turksen & Zhao, 1988) and various FRI techniques (Koczy & Hirota, 1993a, 1993b, 1997) for FIS models. AR and/or FRI techniques are developed from the principles of similarity measure, fuzzy ordering, fuzzy partial ordering, and fuzzy distance. They are introduced as a solution to an incomplete rule base, which allows an unknown rule consequent of an observation to be predicted.

2.4 Data-Driven FIS Models

2.4.1 Definition and Background on Data-Driven FIS Model

The basic configuration of fuzzy rule-based systems are a fuzzification interface, a knowledge base, a decision-making logic and a defuzzification interface, as depicted in Figure 2.3.

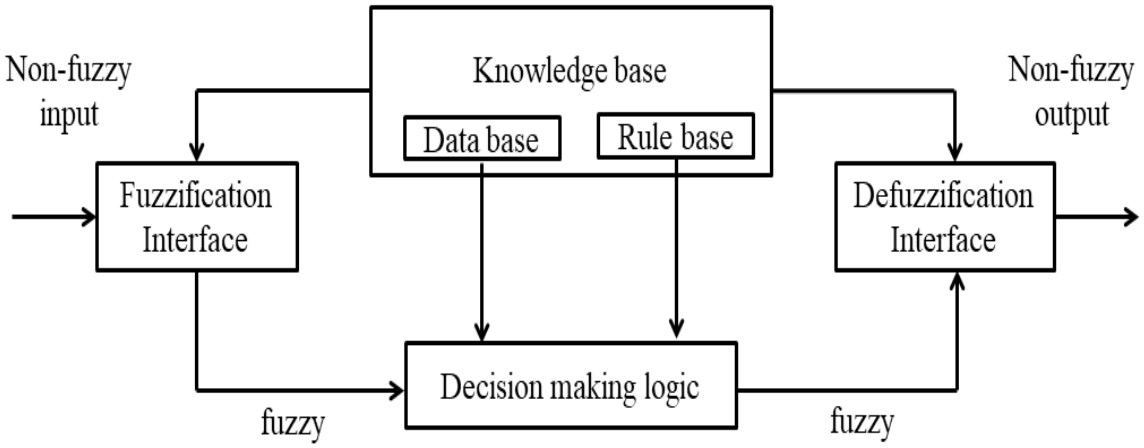

Figure 2.3: Basic configuration of a fuzzy rule-based system

The knowledge base of fuzzy rule-based system consist of two component, i.e., a data base (i.e., fuzzy rule are automatically generating from numerical data) and a rule base (i.e., fuzzy rule are obtained from human experts). There are two factors that determine a data base i.e., a fuzzy partition of an input space and membership functions of antecedent fuzzy sets. The domain interval of the ith input variable x_i is divided into k_i fuzzy sets and labelled as $A_{i,1}, A_{i,2}, \cdots, A_{i,k_i}$ for $i = 1,2, \cdots, n$. Then the n-dimensional input space is divided into $k_1 k_2 \cdots k_n$ fuzzy subspace: $\left(A_{1,j_1}, A_{2,j_2}, \cdots, A_{n,j_n}\right)$, $j_i = 1,2, \cdots, k_i$. For an example, as depicted in Figure 2.4, two dimensional input spaces with the fuzzy subspace, i.e., $\left(A_{1,j_1}, A_{2,j_2}\right)$ is illustrated.

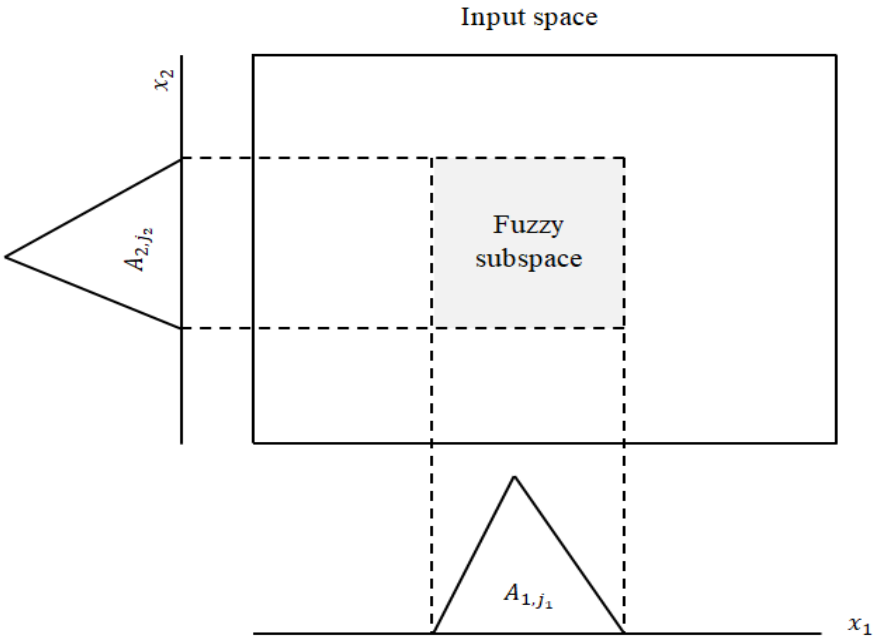

Figure 2.4: Fuzzy subspace

On the other hand, the membership function used for the antecedent fuzzy sets can be in any shape, e.g., triangle-shaped, trapezoid-shaped and bell-shaped. For example, as depicted in Figure 2.5, a fuzzy partition for $k_1 = k_2 = 5$ in the case of two input fuzzy rule-based system.

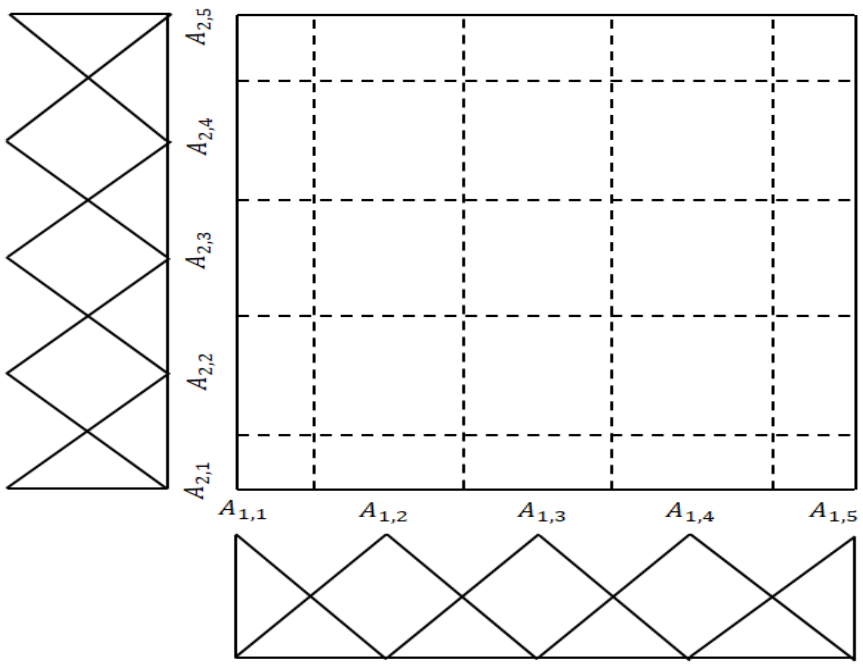

Figure 2.5: Fuzzy partition for $k_1 = k_2 = 5$

2.4.2 Previous Research on the Data-Driven FIS Model

In this section, the related researches on data-driven fuzzy rule induction methods are further discussed. There are three families of data-driven fuzzy rule induction methods, namely *ad hoc* methods, clustering methods, and hybrid methods. In *ad hoc* methods, fuzzy rules are generated from data usually using a heuristic technique. The rule consequents are calculated using a simple mathematical operation, e.g., weighted average. In clustering methods, the training data samples are clustered into homogeneous groups, and a rule is associated with each group. In hybrid methods, usually soft computing techniques, e.g., neuro-fuzzy, genetic algorithm, and system identification techniques are used with an error minimization function.

2.4.2.1 Wang-Mendel (WM) Method

Wang-Mendel (WM) method is an example of *ad hoc* method which is introduced by Wang and Mendel (Wang, 2003; Wang & Mendel, 1992). The number of fuzzy rules is limited by the number of training data pairs. It does not depend on the fuzzy partition resolution level, i.e., the number of fuzzy sets for each input variable. The generation of the fuzzy rule base using the WM method are as following steps:

1. Each input variable is automatically divided into a user defined number of membership fuzzy sets.
2. One fuzzy rule is generated for each data pairs. An importance degree for each of the generated rules is set. The fuzzy set, A_{i,j_i} with the highest *activation degree* is chosen.
3. The redundant rules are eliminated from the set of rules.

4. If two or more rules of the remaining set have equal antecedents, the rule with the smallest importance degree will be eliminated to avoid conflicting rules.

5. Compose the final rule base using the remaining rules.

For example, fuzzy region for two input fuzzy rule-based system using WM method are depicted in Figure 2.6.

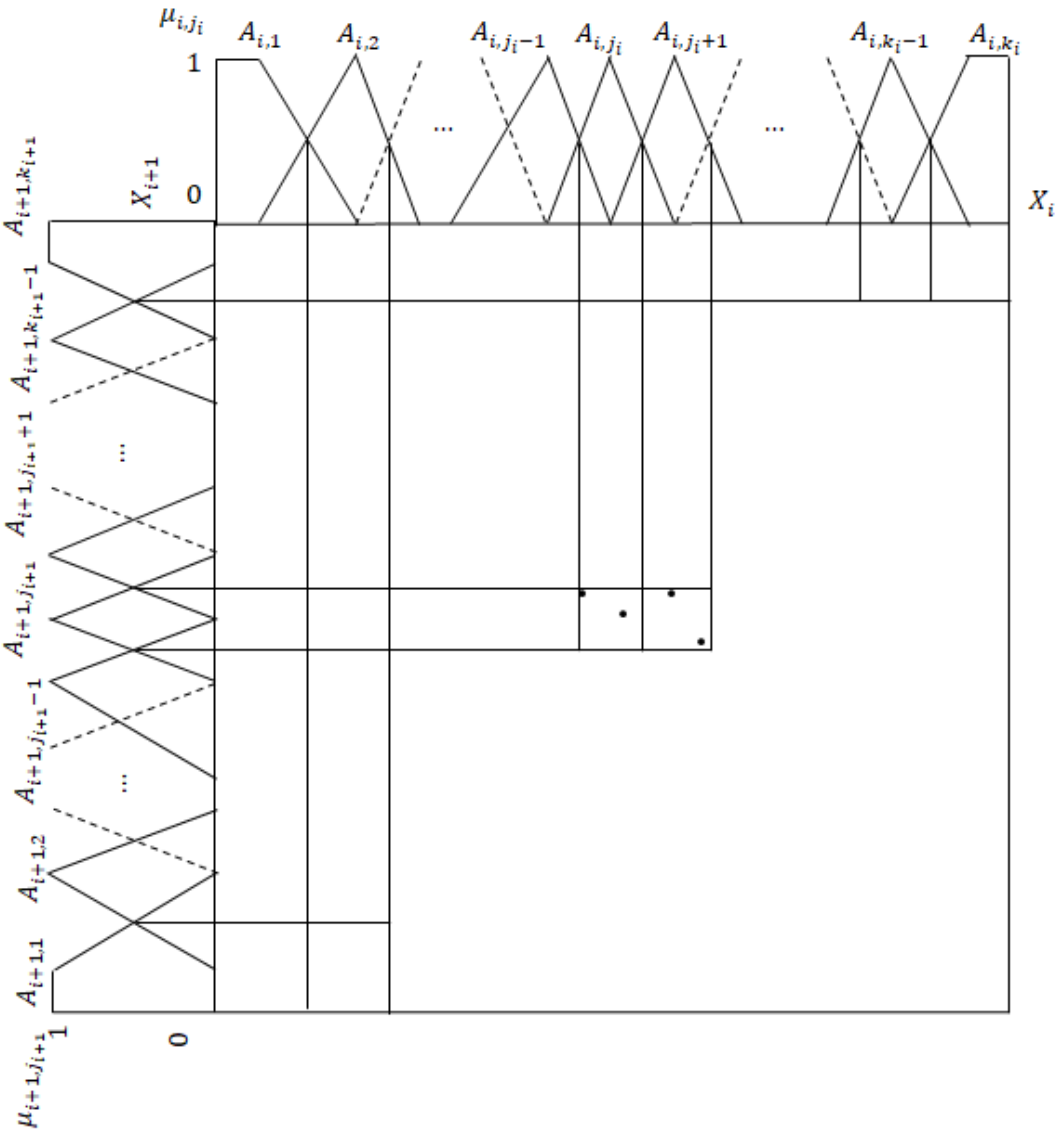

Figure 2.6: Fuzzy region of the fuzzy rule base for the WM method

2.4.2.2 Ishibuchi Method

Ishibuchi method is another example of *ad hoc* method which is introduced by Ishibuchi *et al.* (Ishibuchi et al., 1993; Nozaki et al., 1997). All the fuzzy rules corresponding to the possible combination of the inputs are implemented. Nozaki *et al.* (Nozaki et al., 1997) proposed a simple heuristic method to calculate the rule conclusion, which are real numbers. For example, fuzzy region for two input fuzzy rule-based system using Ishibuchi method are depicted in Figure 2.7.

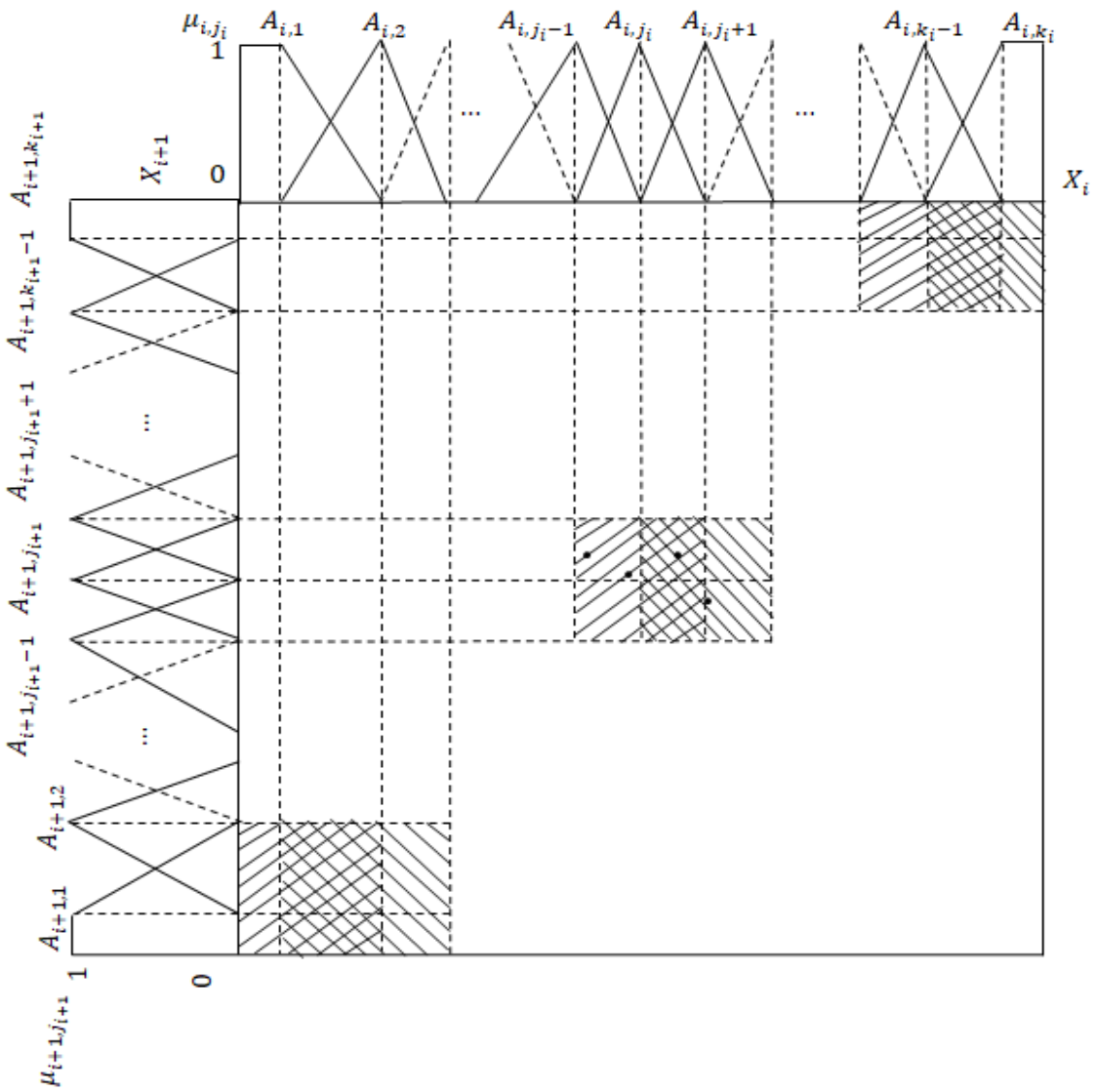

Figure 2.7: Fuzzy region of the fuzzy rule base for the Ishibuchi method

2.4.2.3 Fuzzy *C*-Means Clustering Method (FCMC)

In clustering methods, the training data samples are clustered into homogeneous groups where the space partitioning is derived from the data partitioning. Each cluster is associated with a membership function that expresses the degree to which individual data points belong to the cluster. Then, a rule is associated with each group to each cluster. Among all fuzzy clustering methods, Fuzzy c-Means Clustering (FCMC) method (Dunn, 1973) remains predominant due to its successful application both in the academia as well as industry. FCMC performs clustering by iteratively searching for a set of fuzzy clusters and the associated cluster centres that represent the structure of the data as best as possible. The algorithm relies on the user to specify the number of clusters present in the set of data to be clustered. The algorithm stops when either the error is below a certain tolerance value or its improvement over the previous iteration is below a certain threshold.

2.4.2.4 Adaptive Neuro-Fuzzy Inference System (ANFIS)

Adaptive Neuro-Fuzzy Inference System (ANFIS) is an example of hybrid method which is introduced by Jang *et al.* (Jang et al., 1997). The inference system is implemented in a neuro networks (NN) that uses an hybrid method to adjust the parameters in its nodes, a gradient method and the least squares estimate (LSE) to identify the parameters. ANFIS can construct an input-output mapping based on both expert knowledge (in the form of linguistic rules) and specified input-output data pairs.

For simplicity, two inputs and one output FIS model is considered, as illustrated in Figure 2.8. The node functions in the same layer belong the same function family as described below:

Layer 1: Parameters in this layer are referred to as premise parameters. In fact, any differentiable function, membership functions are valid for the nodes in this layer. Every node in this layer is a square node (i.e., adaptive node) with a node function.

Layer 2: Every node in this layer is a fixed node, whose output is the product of all the incoming signal. Each node output represents the firing strength of a rule.

Layer 3: Every node in this layer is a fixed node. The node will calculate the ratio of the rule's firing strength to the sum of all rule's firing strengths.

Layer 4: Every node in this layer is an adaptive node.

Layer 5: The single node in this layer computes the overall output as the sum of all incoming signals.

$$overall\ output = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

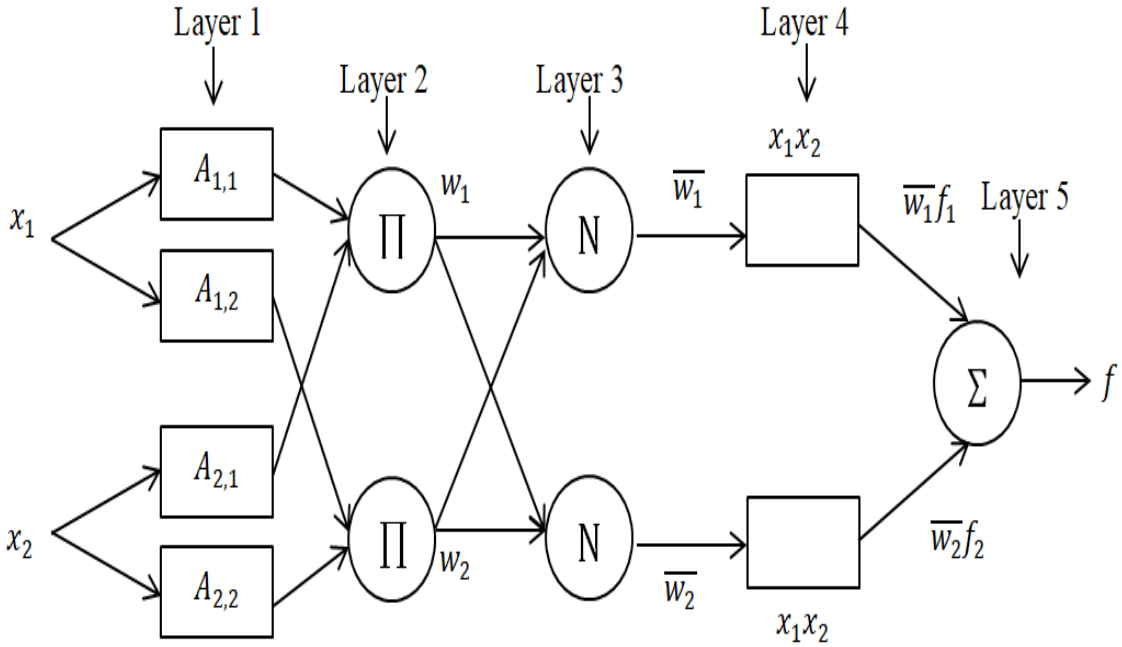

Figure 2.8: ANFIS architecture

2.5 Background and Review on Fuzzy Rule Interpolation (FRI)

2.5.1 Background

In FIS modelling, recent advances are focused on the use of FRI as a solution to an incomplete rule base (Koczy & Hirota, 1993a, 1993b, 1997). Obtaining a complete rule base is sometimes not feasible, especially for multi-attribute FIS models, owing to the large number of rules required (Koczy & Hirota, 1993a, 1993b, 1997). This phenomenon is known as the "curse of dimensionality", which indicates the degree of exponential increase in the number of rules of an FIS model when extra number of inputs are added (Jang et al., 1997; Jin, 2000; Lin & Lee, 1995). There are several reasons for an incomplete rule base to occur, which include (i) the system developer does not have full knowledge of the process that he/she tries to model, or it is difficult to solicit a large number of rules from the domain experts (Jenei, 2001; Jenei, Klement, & Konzel, 2002; Wong, Tikk, Gedeon, & Kóczy, 2005); (ii) the removal of redundant fuzzy rules in order to reduce the complexity of the modelled problem (Wong et al., 2005); (iii) the data samples collected do not sufficiently represent the input space for constructing an FIS model with machine learning techniques, e.g. neural or evolutionary methods (Jenei, 2001; Jenei et al., 2002; Wong et al., 2005).

As such, a sparse rule base is normally introduced to describe an FIS model when full knowledge (or a complete rule base) of the problem is not available (Jang et al., 1997; Koczy & Hirota, 1993a, 1993b, 1997). When it has one or more unknown consequents, the rule base contains "gaps". Conventional FIS models, e.g. the Mamdani and Takagi-Sugeno-Kang models, map a rule antecedent (observation) in the input domain to zero if its rule consequent (conclusion) is undefined. However, this assumption may not be true or

appropriate always; and this is the so-called “tomato classification problem” (Koczy & Hirota, 1993a; Sánchez & Zadeh, 1987).

2.5.2 Review on Fuzzy Rule Interpolation (FRI)

Turksen and Zhao proposed an Approximate Analogical Reasoning Scheme (AARS) to predict the rule consequent associated with an empty antecedent based on the degree of overlapping (as a similarity measure) between the consequent and the antecedents of other existing rules. As depicted in Figure 2.9, two existing rules, Rule #i ($A_i \rightarrow B_i$) and Rule #j ($A_j \rightarrow B_j$), are provided. The AARS is used to predict the consequent of observations $A^{\#}$, $B^{\#}$. From Rules #i and #j, ${B_i}^{\#}$ and ${B_j}^{\#}$ are fired, respectively. ${B_i}^{\#}$ and ${B_j}^{\#}$ are aggregated to form a resulting fuzzy set, $B^{\#}$.

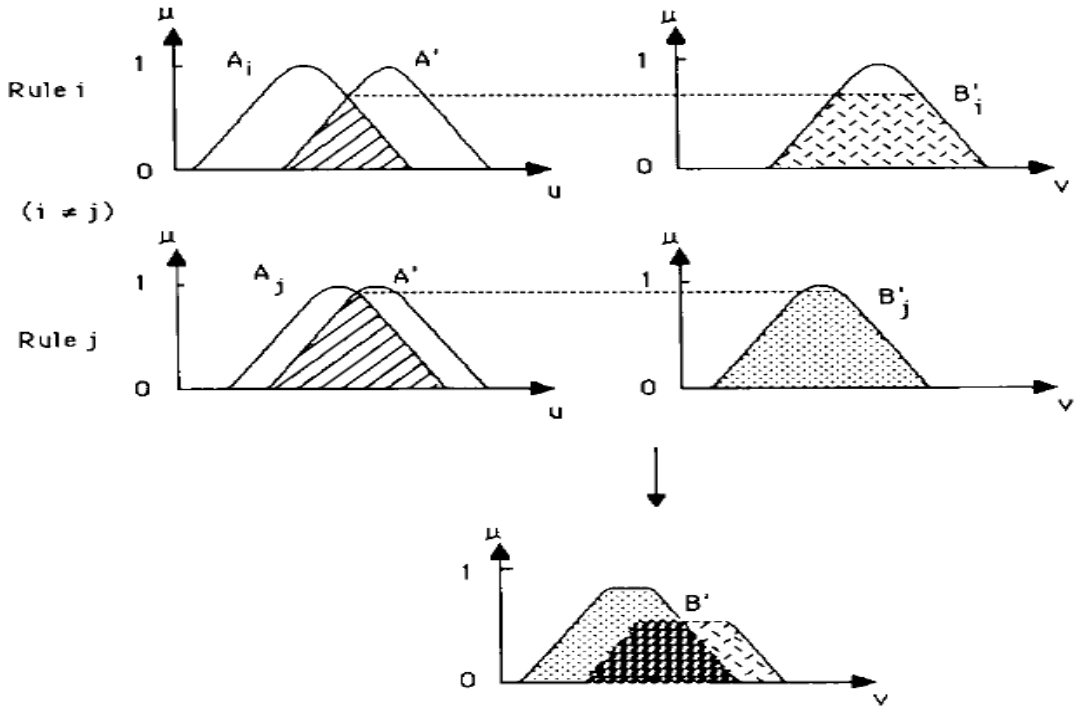

Figure 2.9: The procedure of the AARS (adapted from Turksen, and Zhao, 1988)

A gradual inference rule with the principle "the more an observation to a rule antecedent, the more the conclusion should be to the rule consequent corresponding to the given rule antecedent" was proposed by Dubois and Prade (1992). This approach is viewed as an improvement of the AR scheme by Wong *et al.* (2005). Various FRI techniques based on the concept of *Fundamental Equation of Rule Interpolation* (FERI) are further proposed by Wong *et al.* (2005). FERI describes the connection of the distance ratio between a rule consequent and other rule antecedents. Figure 2.10 shows an example of FERI for an FRI technique known as solid-cut FRI. In this example, two existing rules, i.e., R_1 $(A_1 \rightarrow B_1)$ and R_2 $(A_2 \rightarrow B_2)$ are provided. The FRI technique is used to predict the consequent of observations A^i, B^i. Assume that the distance ratio between fuzzy sets at antecedent and consequent are identical, as suggested by FERI, B^i can be predicted with a linear interpolation technique.

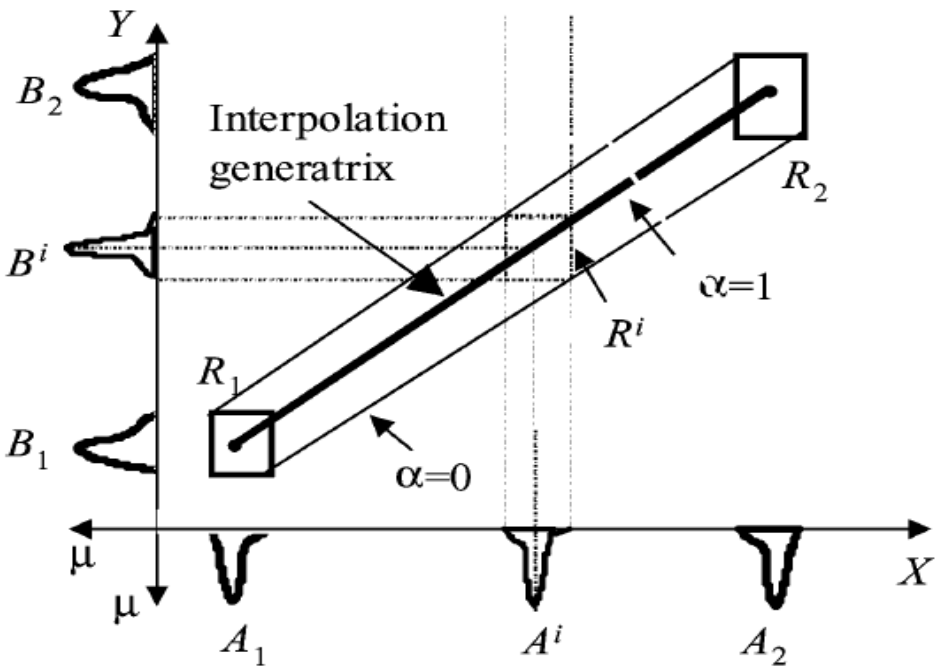

Figure 2.10: Basic concept of FERI with solid-cut FRI (adapted from (Baranyi, Kóczy, & Gedeon, 2004))

Comparing with conventional AARSs (Turksen & Zhao, 1988), FRI utilizes a *distance function*, which can be viewed as a type of similarity measure function (Raha, Pal, & Ray, 2002). Over the years, various FRI techniques have been developed. Examples include Kóczy and Hirota (KH) FRI (Koczy & Hirota, 1993a, 1993b, 1997), a solid cut-based FRI (Baranyi et al., 2004), a modified α-cut-based interpolation (MACI) (Tikk & Baranyi, 2000), Huang and Shen (HS) HS FRI (Z. Huang & Shen, 2006), an interpolation with a function space representation of membership functions (Yam, Wong, & Baranyi, 2006), an α-cut and transformation-based FRI (Ko & Chen, 2007), and an area-based FRI (Chang, Chen, & Liau, 2008).

2.6 Monotone FIS Model

2.6.1 Background

Consider a function, $y = f(\boldsymbol{x})$ which fulfils the condition of monotonicity between its outputs, y, with respect to its input i-th input, x_i within the universal of discourse. The relationship of monotonicity properties are presented in Equation 2.8 and Equation 2.9. The output of model either monotonically increases or decreases as x_i increases, respectively.

$$f(x_1, x_2, \cdots, x_i^1, \cdots, x_n) \leq f(x_1, x_2, \cdots, x_i^2, \cdots, x_n) \tag{2.8}$$

$$f(x_1, x_2, \cdots, x_i^1, \cdots, x_n) \geq f(x_1, x_2, \cdots, x_i^2, \cdots, x_n) \tag{2.9}$$

where, $x_i^1 < x_i^2$. The monotonicity property is an additional qualitative information or knowledge that can be exploited to obtain an interpretable and optimized FIS model (Tay & Lim, 2011c).

2.6.2 Previous Researches on the Monotonicity Property in FIS

In this section, the related researches on the monotonicity property in FIS model are further discussed.

2.6.2.1 Finding from H.Seki, H.Ishii and M.Mizumoto' works (Seki et al., 2010)

The authors pointed out that even if the consequent parts in fuzzy rules are monotone, the inference results are not necessarily a monotone when the antecedent parts are non-comparable fuzzy set. In this paper, they suggested a method (functional-type of Single Input Rule Model (SIRM) approach) to obtain the condition for the monotonicity inference results. In addition, the monotonicity condition for two input data can be simplified by using the TSK FIS model.

The suggested conditions (*sufficient conditions*) by the author in monitoring the monotonicity property in the result of fuzzy inference method are shown in Equation 2.10 and Equation 2.11.

$$Rules1 = \begin{cases} x_1 = A_{1,1} \rightarrow y_1 = x_1 + 0.1 \\ x_1 = A_{1,2} \rightarrow y_1 = x_1 + 0.2 \\ x_1 = A_{1,3} \rightarrow y_1 = x_1 + 10 \end{cases} \tag{2.10}$$

$$Rules2 = \begin{cases} x_2 = A_{2,1} \rightarrow y_2 = x_2 + 0.3 \\ x_2 = A_{2,2} \rightarrow y_2 = x_2 + 0.5 \end{cases} \tag{2.11}$$

where, fuzzy set $A_{1,1}, A_{1,2}, A_{1,3}$ of *Rules-1* are non-comparable fuzzy set, and the fuzzy set $A_{2,1}, A_{2,1},$ of *Rules-2* are comparable fuzzy set that fulfill the monotonicity increasing function. The consequent part in the *Rule-1* and *Rule-2* satisfy the monotonicity condition

of $f_{i,1}(x_i) \leq f_{i,2}(x_i) \leq \cdots \leq f_{i,k_i}(x_i)$. For that reason, even if the monotonicity conditions are satisfies for the consequent parts in the functional type SIRM method and TSK inference method, the inference result derived with these inference methods are not necessarily a monotone system especially when the antecedent parts are non-comparable fuzzy set.

2.6.2.2 Finding from K.M. Tay and C.P. Lim' s works (Tay & Lim, 2011b)

The authors have introduced an assessment model (mathematical model) that produced by a measuring index. From the numerical score in the assessment model, a decision can be made and an action can be taken. The *sufficient conditions* of the FIS model with Gaussian membership function for a fuzzy assessment model to fulfill the monotonicity output property using a derivative approach is introduced by the authors. For the *sufficient condition*, the authors assume that all the component in the weighted addition series are always *greater than or equal to zero,* or *less than or equal to zero*. The general condition can be simplified into two conditions: the antecedent part and the consequent part (Tay & Lim, 2011a):

Condition 1: At the rule antecedent part,

$$(d\mu_i^p(x)/dx\ )\ /\mu_i^p(x)\ \geq (d\mu_i^q(x)/dx\ )\ /\mu_i^q(x) \qquad :p > q \tag{2.12}$$

$(d\mu(x)/dx\ )\ /\ \mu(\ x)$ is the ratio between the rate of change in the membership degree and the membership degree itself. The derivative of a Gaussian Membership Function (MF) with respect to *x* is:

$$G'(x) = -((x - c)/\sigma^2)\, G(x) \tag{2.13}$$

where, c is representing the center of an Gaussian MF, and σ is representing the width length of an Gaussian MF. The Gaussian MF condition is then derived in$(G'(x)/\, G(x))$, where a linear equation can be obtained as in Equation 2.14 (Tay & Lim, 2011b).

$$E(x) \;= G'(x)/\, G(x) \;=\; -\,(1/\sigma^2)\, x + \;(c/\,\sigma^2) \tag{2.14}$$

Condition 2: At the rule consequent part,

$$b_{j_1,j_2,\cdots,j_{i=p},\cdots,j_n} - b_{j_1,j_2,\cdots,j_{i=q},\cdots,j_n} \geq 0 \tag{2.15}$$

$$b_{j_1,j_2,\cdots,j_{i=p},\cdots,j_n} - b_{j_1,j_2,\cdots,j_{i=q},\cdots,j_n} \leq 0 \tag{2.16}$$

where Equation 2.15 is for $dy/dx_i \geq 0$ and Equation 2.16 is for $dy/dx_i \leq 0$ respectively. This condition suggests that a monotonically ordered rule base is required (Tay & Lim, 2011b).

2.6.2.3 Finding from J. M. Won, S. Y. Park and J. S. Lee's works (Won et al., 2002)

The authors have presented the *sufficient conditions* on the parameter of the TSK fuzzy system under which the output of the TSK fuzzy system becomes monotonicity with respect to its input. The authors have reported that many real-world engineering systems satisfy the monotonicity property. Two examples, i.e., the cart-pole system and the magnetic crane controller system, are explained by the authors. They suggested that FIS models that preserve the monotonicity property are able to perform better in an

approximate solution rather than the actual control mechanism. They pointed that the derived monotonicity *sufficient conditions* consists of two parts: the conditions on the consequent parts and the conditions on the fuzzy membership functions.

2.6.2.4 Finding from Broekhoven and De Baets' works (Broekhoven & Baets, 2009)

The authors pointed out that it is important to fully exploit the monotonicity property as additional qualitative information, especially in the case whereby the number of data samples is small (Broekhoven & Baets, 2009).

Even though the *sufficient conditions* (Won et al., 2002) are important and useful, but they do not explain the scenario if the min operator is used as the AND operator. Thus, they further investigates on the use of the three basic AND operators, i.e., minimum, product, and Łukasiewicz, in monotonicity-preserving FIS models. A guideline to design monotonicity-preserving FIS models with minimum, product, and Łukasiewicz operators is provided. This was an attempt to go beyond the boundary of the *sufficient conditions*, and it has been shown that the choices of mathematical operators and the properties of the output membership functions are critical.

2.6.2.5 Finding from P. Lindskog and L. Ljung's works (Lindskog & Ljung, 2000)

The authors have pointed out the importance of the monotonicity property in FIS-based control problems. They focused on FIS models with triangular membership functions. It was further assumed that the triangular membership functions are orthogonal, i.e., summation of the membership value at every point of the input space is equal to1.

A procedure to construct a monotonicity-preserving FIS model was suggested by the authors. An FIS model structure that ensures the input–output monotonicity is

proposed, which is used to identify the dynamic system whose output is monotonicity with respect to its input. They further parameterized the FIS structure. Besides that, the problems that related to constrains for each parameter is developed and imposed in the FIS designing process.

2.6.2.6 Finding from Zhao and Zhu's works (Zhao & Zhu, 2000)

The authors suggested that for most process control problems, regardless of single-input single-output, or multi-input multi-output problems, the relationship of the input and output is obeying the monotonicity property, i.e., the output of the process can be expressed as a monotonicity function of the input variables. They pointed out that the monotonicity property is important to ensure the stability and the steady state error of an FIS-based control problem. The conditions for single-attribute and two- attribute Mamdani FIS models to be of monotonicity are shown with the FIS operations analyzed step-by-step. As a conclusion, the authors' suggestion is that as long as the rule base is monotonically-ordered, a single- attribute Mamdani fuzzy model can be of monotonicity, and a two-attribute Mamdani fuzzy model can be roughly of monotonicity.

2.6.2.7 Finding from Wu and Sung's works (Wu & Sung, 1996)

The authors have explained another research that related to the monotonicity property. They suggested that the monotonicity property is important for the stability analysis of FIS-based control problem. They also stress the importance of the monotonicity property for JPEG models in image compression. They focus in FIS with triangular membership functions. The author has proposed a new defuzzification operator, i.e., mean-of-inversion (MOI) for monotonicity-preserving FIS models. The function of

MOI operator is to defuzzifier each rule separately, instead of superimposing all fired rules before defuzzification.

2.6.2.7 Finding from A. Restrepo's work (Restrepo & Bovik, 1993)

The author has pointed out that concept of local monotonicity is relatively unexplored in FIS filed and has been mostly limited by the theoretical prove of a class of non-linear filter that includes, median filter, rank-order filter and order static filter. The concept of local monotonicity appears in the study of the set of root signals of the median filter (sub-optimal smoother under the measurement) and provides a measure of the smoothness of a signal.

Local monotonicity regression is a direct (one-dimensional signal) approach that used to obtain the optimal result for a smooth discrete signal under a local monotonicity standard (Restrepo & Bovik, 1993). Therefore, it is often used as a standard technique to measure the smoothness of a signal, since it has a limit that restricts the changes of the trend (increasing to decreasing and vice versa).

The local monotonicity can use as an indicator to point out the contamination of a signal with noise. When given a non-local monotonicity signal, it has the capability to find the signal that is both similar to the given signal and locally monotonic (Restrepo & Bovik, 1993).

2.7 Summary

In this chapter, the background and literature reviews on fuzzy set theory, fuzzy production rule, fuzzy reasoning, FIS and FRI techniques are presented. The literature reviews are focused on FIS models and the monotonicity property.

In the next chapter, an analysis of the zero-order TSK FIS model is provided. A *sufficient conditions* for the zero-order TSK FIS model is established. A monotone test is established to evaluate the monotonicity property of an FIS model.

CHAPTER 3

AN ANALYSIS OF THE MONOTONE ZERO-ORDER TSK FIS MODEL

3.1 Chapter Overview

In the previous chapter, literature review was presented. In this chapter, an analysis of the zero-order TSK FIS model is provided. A *sufficient conditions* for the zero-order TSK FIS model is established. A monotone test is established to evaluate the monotonicity property of an FIS model.

In Section 3.2, number definitions are described. In Section 3.3, an analysis of the zero-order TSK FIS model is presented. It is shown that with convex and normal trapezoidal/triangular membership functions, coupled with a strong fuzzy partition strategy, a zero-order TSK FIS model always satisfy the monotonicity property, if the fuzzy rules are monotone and complete. This result a useful *sufficient conditions.* In Section 3.4, a monotonicity test to evaluate the monotonicity property of an FIS model is developed. In Section 3.5, two benchmark examples, i.e., an FIS-based occurrence model (a single-attribute example) for Failure Mode and Effect Analysis (FMEA) and an FIS-based risk model (a multi-attribute example) are presented to illustrate the usefulness of the *sufficient conditions.* Finally, concluding remarks are given in Section 3.6.

3.2 Definitions

Definition 3.1: Consider an n-input space, X, and an output space, Y.

3.1.1 Vector variable $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ where $x_i \in X_i$, $i = 1, \cdots, n$, and scalar variable y are the set elements of $X = X_1 \times X_2 \times \cdots \times X_n$ and Y, i.e., $\boldsymbol{x} \in X$ and $y \in Y$.

3.1.2 Integer vector $\boldsymbol{j} = (j_1, j_2, \cdots, j_n)$, where $j_i = 1,2, \cdots, k_i$ denotes some specified numbers $k_i \in \mathbb{N}_{>0}$.

3.1.3 An n-dimensional unit vector in direction i is denoted $\boldsymbol{e}_i = (0, \cdots, 1, \cdots 0)$, e.g. vector $(j_1, \cdots, j_i + 1, \cdots, j_n)$ is written as $\boldsymbol{j} + \boldsymbol{e}_i$.

3.1.4 The lower and upper bounds of X_i are denoted $\underline{x_i}$ and $\overline{x_i}$, respectively.

Definition 3.2: A fuzzy set A on the X_i domain is a normal trapezoidal fuzzy set, denoted $A = (a, b, c, d)$, if its membership function is given by (see Figure 3.1):

$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \le x \le b, \\ 1, & b \le x \le c, \\ \dfrac{d-x}{d-c}, & c \le x \le d, \\ 0. & otherwise. \end{cases} \tag{3.1}$$

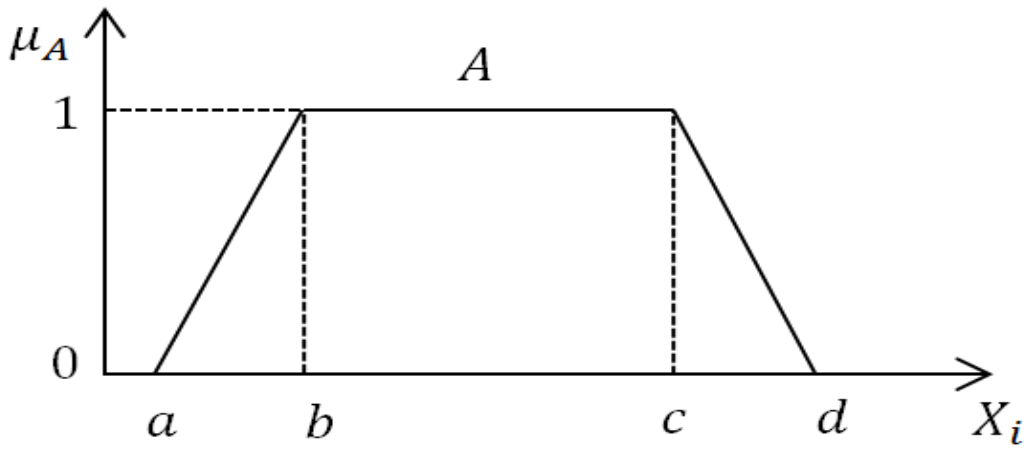

Figure 3.1: The membership function of a trapezoidal fuzzy set

If $b = c$, then A is a normal triangular fuzzy set.

Definition 3.3: Domain X_i of input variable x_i is divided into k_i normal trapezoidal fuzzy sets, denoted $A_{i,j} = (a_{i,j}, b_{i,j}, c_{i,j}, d_{i,j}), j = 1, \cdots, k_i$, with the corresponding membership functions, $\mu_{i,j}(x)$.

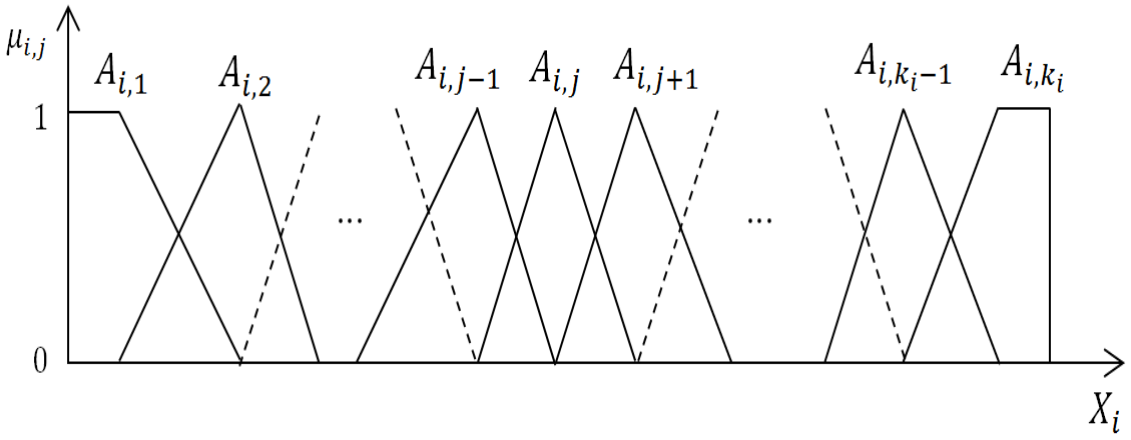

Figure 3.2: Fuzzy sets of k_i partitions

3.3.1 $\sum_{j=1}^{k_i} \mu_{i,j}(x) = 1$ is always true (i.e., a strong partition).

3.3.2 $\mu_{i,1}(\underline{x_i}) = 1$ and $\mu_{i,k_i}(\overline{x_i}) = 1$.

3.3.3 $A_{i,j} \leq A_{i,j+1}$, where $j = 1,2, \cdots, k_i - 1$ is always true. Note that $A_{i,j} \leq A_{i,j+1}$ is equivalent to the following simultaneous conditions, i.e., $a_{i,j} \leq a_{i,j+1}, b_{i,j} \leq b_{i,j+1}, c_{i,j} \leq c_{i,j+1}$ and $d_{i,j} \leq d_{i,j+1}$ (Won et al., 2002).

3.3.4 In particular, for any pair of consecutive fuzzy sets belonging to the partition of the X_i space, we have $a_{i,j+1} = c_{i,j}$ and $b_{i,j+1} = d_{i,j}$. The first equality is shown in Figure 3.2 for the first two fuzzy sets, $A_{i,1}$ and $A_{i,2}$, and the second equality holds for the last two fuzzy sets, A_{i,k_i-1} and A_{i,k_i}.

Definition 3.4: An antecedent of the $X_1 \times X_2 \times \cdots \times X_n$ space, denoted $A_{1,j_1} \wedge \cdots \wedge A_{n,j_n}$, is simplified as $A_{\boldsymbol{j}}$ where $\boldsymbol{j} = (j_1, j_2, \cdots, j_n)$. Each antecedent is mapped to a real value on the Y space, i.e., $b_{\boldsymbol{j}}$. A fuzzy rule which maps the input space to the output space is denoted $R_{\boldsymbol{j}}: A_{\boldsymbol{j}} \rightarrow b_{\boldsymbol{j}}$. The collection of $R_{\boldsymbol{j}}$ for all integer vectors $\boldsymbol{j}$, where $j_i = 1,2, \cdots, k_i$, forms the fuzzy rule base.

3.4.1 $R_{\boldsymbol{j}}: A_{\boldsymbol{j}} \rightarrow b_{\boldsymbol{j}}$ is interpreted as $IF\ \left(x_1\ is\ A_{1,j_1}\right)\ AND\ \cdots AND \left(x_n\ is\ A_{n,j_n}\right), THEN\ \left(y = b_{j_1,\cdots,j_n} = b_{\boldsymbol{j}}\right)$.

3.4.2 The fuzzy rule base is *complete* if all $b_{\boldsymbol{j}}$ are known. Otherwise, the fuzzy rule base is *incomplete*. A *complete* fuzzy rule base should have M fuzzy rules, i.e., $M = \prod_{i=1}^{n} k_i$.

3.4.3 The fuzzy rule base is *monotone* if $b_{\boldsymbol{j}+\boldsymbol{e}_i} \geq b_{\boldsymbol{j}}$ for all $i = 1, \cdots, n$ and $\boldsymbol{j}$ such that $j_i < k_i$. Otherwise, the fuzzy rule base is *non-monotone*.

Without loss of generality, we only consider a *monotone* increasing rule base in this paper.

3.3 An Analysis of the Zero-Order TSK FIS Model

A zero-order TSK FIS model with n-input, i.e., $y = f(\boldsymbol{x})$, is considered. The output, y, is obtained as follows.

$$y = f(\boldsymbol{x}) = \frac{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \left(\mu_{\boldsymbol{j}}(\boldsymbol{x}) \times b_{\boldsymbol{j}}\right)}{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \mu_{\boldsymbol{j}}(\boldsymbol{x})} \tag{3.2}$$

where $\mu_{\boldsymbol{j}}(\boldsymbol{x}) = \mu_{j_1,\cdots,j_n}(\boldsymbol{x}) = \prod_{i=1}^{n} \mu_{i,j_i}(x_i)$ is the *activation degree (firing strength)* of rule $R_{\boldsymbol{j}}$ and $\boldsymbol{j} = (j_1, \cdots, j_n)$ is the corresponding combination of the input fuzzy sets.

3.3.1 The Sufficient Conditions

Remark 3.1: The denominator in Equation 3.2 always equals to 1 (see Definition 3.3.1), which reduces Equation 3.2 to 3.3, as follows (Ishibuchi et al., 1994 and Lindskog et al., 2000).

$$y = f(\boldsymbol{x}) = \sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \left(\mu_{\boldsymbol{j}}(\boldsymbol{x}) \times b_{\boldsymbol{j}}\right) \tag{3.3}$$

Definition 3.5: The zero-order TSK FIS model is *monotone* with respect to x_i, if $f(\boldsymbol{x}) \geq f(\boldsymbol{z})$ for all n-vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ of X such that $x_i \geq z_i$.

Proposition 3.1. Suppose that the zero-order TSK FIS model has trapezoidal/ triangular fuzzy sets fulfilling the conditions of Definition 3.3, and has a *complete* and *monotone* rule base (see Definition 3.4.2 and 3.4.3). As such, $y = f(\boldsymbol{x})$ is a monotonically increasing function in accordance with Definition 3.5.

Proof. Equation 3.3 is considered. The output, y, is re-arranged, as follows.

$y = f(\boldsymbol{x}) = \sum_{j_1=1}^{k_1} \cdots \sum_{j_{n-1}=1}^{k_{n-1}} \prod_{i=1}^{n-1} \mu_{i,j_i+\boldsymbol{e}_i}(x_i) \left[\sum_{j_i=1}^{k_i} \mu_{i,j_i}(x_i) \times b_{\boldsymbol{j}}\right]$. Note that, $\sum_{j_1=1}^{k_1} \cdots \sum_{j_{n-1}=1}^{k_{n-1}} \prod_{i=1}^{n-1} \mu_{i,j_i+\boldsymbol{e}_i}(x_i)$ is considered as a *constant* and $\sum_{j_i=1}^{k_i} \mu_{i,j_i}(x_i) \times b_{\boldsymbol{j}}$ is considered as a *variable*. To simplify the notation, $y_i(x_i) = \sum_{j_i=1}^{k_i} \mu_{i,j_i}(x_i) \times b_{\boldsymbol{j}}$ is denoted. The following two scenarios are considered.

Scenario A: Consider any two fuzzy sets at X_i, i.e., $\mu_{i,j_i}(x_i)$ and $\mu_{i,j_i+1}(x_i)$ for interval $\left[a_{i,j_i+1} = c_{i,j_i}, b_{i,j_i+1} = d_{i,j_i}\right] \in \left[\underline{x_i}, \overline{x_i}\right]$ (see Definition 3.3), and $b_{\boldsymbol{j}+\boldsymbol{e}_i} \geq b_{\boldsymbol{j}}$. We have

$$y_i(x_i) = \mu_{i,j_i}(x_i) \times b_{\boldsymbol{j}} + \mu_{i,j_i+1}(x_i) \times b_{\boldsymbol{j}+\boldsymbol{e}_i}$$

$$= \left[1 - \mu_{i,j_i+1}(x_i)\right] b_{\boldsymbol{j}} + \mu_{i,j_i+1}(x_i) b_{\boldsymbol{j}+\boldsymbol{e}_i}$$

$$= b_{\boldsymbol{j}} - \mu_{i,j_i+1}(x_i) b_{\boldsymbol{j}} + \mu_{i,j_i+1}(x_i) b_{\boldsymbol{j}+\boldsymbol{e}_i}$$

$$= \left[b_{\boldsymbol{j}+\boldsymbol{e}_i} - b_{\boldsymbol{j}}\right] \mu_{i,j_i+1}(x_i) + b_{\boldsymbol{j}}$$

If $b_{\boldsymbol{j}+\boldsymbol{e}_i} \geq b_{\boldsymbol{j}}$ is true, and $\mu_{i,j_i+1}(x_i)$ is an increasing function on $\left[a_{i,j_i+1} = c_{i,j_i}, b_{i,j_i+1} = d_{i,j_i}\right]$, then $y_i(x_i)$ is an increasing function (i.e., non-decreasing) on that interval.

Scenario B: Consider for interval $\left[b_{i,j_i} = d_{i,j_i-1}, a_{i,j_i+1} = c_{i,j_i}\right] \in \left[\underline{x_i}, \overline{x_i}\right]$, any two fuzzy sets at X_i, i.e., $\mu_{i,j_i}(x_i) = 1$ and $\mu_{i,j_i+1}(x_i) = 0$, where $j_i = 1,2,\cdots,k_i - 1$, are always true. We have

$$y_i(x_i) = \mu_{i,j_i}(x_i) \times b_{\boldsymbol{j}} + \mu_{i,j_i+1}(x_i) \times b_{\boldsymbol{j}+\boldsymbol{e}_i}$$

$$y_i(x_i) = b_{\boldsymbol{j}}$$

Notice that $y_i(x_i)$ remains stationary (i.e., non-decreasing) for interval $\left[b_{i,j_i} = d_{i,j_i-1}, a_{i,j_i+1} = c_{i,j_i}\right]$. Combining both Scenario (A) and Scenario (B), $y_i(x_i)$ is an increasing function (i.e., non-decreasing) for $\left[\underline{x_i}, \overline{x_i}\right]$ if the fuzzy rules are *monotone*. We

can conclude that the zero-order TSK FIS model is always *monotone* if the fuzzy rules are *monotone* and *complete* (Definition 3.4).

Proposition 3.1 is useful as a set of *sufficient conditions* to develop knowledge-driven *monotone* FIS models. Convex and normal trapezoidal and/or triangular fuzzy sets with a strong fuzzy partition strategy are used as a guideline to design fuzzy MFs of a zero-order TSK FIS model. The fuzzy rules from human experts should be *complete* and *monotone*, which are useful for assessment and decision making problems (Pang, Tay, & Lim, 2016) e.g., fuzzy FMEA (Jee, Tay, & Lim, 2015; Kerk, Tay, & Lim, 2017). Note that our previous studies focus on the use of Gaussian fuzzy sets (Pang et al., 2016). In the following chapter, **Proposition 3.1** acts as a mathematical condition for modelling *data-driven monotone FIS models*.

3.4 The Monotonicity Test

The monotonicity property of the zero-order TSK FIS model can be measured by comparing the neighbouring outputs in pairs from an FIS model using a monotonicity test (Hennie Daniels & Velikova, 2010; Goldreich, Goldwasser, Lehman, Ron, & Samorodnitsky, 2000). The aim is to represent the *monotone* relationship using a numerical value from 0 to 1. The procedure of the monotonicity test is as follows.

A. Let $X_i = \left[\underline{x_i}, \overline{x_i}\right]$, $i = 1, \cdots, n$.

B. Take $N_i + 1$ equally spaced points $x_{i,m}$ in each interval X_i:

$x_{i,m} = \underline{x_i} + m\delta_i$, $m = 0, \cdots, N_i$ where $\delta_i = \left(\overline{x_i} - \underline{x_i}\right)/N_i$ is a fixed step size. Note that $x_{i,m} < x_{i,m+1} \leq \overline{x_i}$ for all $m < N_i$ and $x_{i,N_i} = \overline{x_i}$.

C. Evaluate $f(\boldsymbol{x})$ for all combinations of grid points $\boldsymbol{x} = (x_{1,m_1}, \cdots, x_{n,m_n})$ for $m_i \leq N_i$. Compare the values of f for *all neighbouring pairs*, i.e., $f(\boldsymbol{x})$ and $f(\boldsymbol{x} + \delta_k \boldsymbol{e}_k)$ for all indices $k = 1, \cdots, n$ such that $x_{k,m_k} < \overline{x_k}$. For any given pair of grid points $\boldsymbol{x}$ and $\boldsymbol{x} + \delta_k \boldsymbol{e}_k$, compute

$$\text{monotone}(f; \boldsymbol{x}; k) = \begin{cases} 1, & f(\boldsymbol{x} + \delta_k \boldsymbol{e}_k) \geq f(\boldsymbol{x}) \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

There are N_i pairs of neighboring grid points in X_i, i.e., pairs (x_{i,m_i}, x_{i,m_i+1}) for $m_i = 0, \cdots, N_i - 1$. Consequently, there are a total of $\prod_{i=1}^{n}(N_i + 1) - 1$ $\boldsymbol{x}$ for the entire state space X to be evaluated. There is no *neighbouring pair* for $\boldsymbol{x} = (x_{1,N_1}, \cdots, x_{n,N_n})$ to be evaluated.

D. Obtain the degree of monotonicity (DOM) for an FIS model using Equation 3.5:

$$\text{DOM}(f; X; N_1, \cdots, N_n) = \frac{\sum_{\substack{\text{all grid vectors } \boldsymbol{x} \text{ such that} \\ x_i = x_{i,m_i}}} \left[\sum_{\substack{\text{all components } k=1,\cdots,n \text{ of vector } \boldsymbol{x} \\ \text{such that } x_k < \overline{x_k}}} \text{monotone}(f; \boldsymbol{x}; k) \right]}{\prod_{i=1}^{n}(N_i + 1) - 1} \tag{3.5}$$

Remark 3.2*:* If $f(X)$ is *monotone*, $\text{DOM} = 1$.

3.5 Benchmark Examples

3.5.1 The FIS-based Occurrence Model

FMEA is a popular problem prevention methodology that can be interfaced with many engineering and reliability approaches (Stamatis, 2003). The objective of FMEA is

to define, recognize, prioritize, and abolish known and/or potential failures prior to their occurrences in a production process (Stamatis, 2003). FMEA uses a Risk Priority Number (RPN) model to evaluate the risk associated with each failure mode. The RPN score is obtained by multiplying three risk factors, i.e., Severity, Occurrence, and Detection (Hoffmann, 2001), as shown in Equation 3.6.

$$RPN = Severity \times Occurrence \times Detect \tag{3.6}$$

Severity and Occurrence are the frequency and effect of a failure mode, and detection is the effectiveness of the existing control measures to detect a failure mode before it reaches to the customer, subsystem, or component (Hoffmann, 2001). In general, the three risk factors are estimated by experts and the scale table are ranging from "1" to "10" based on commonly agreed evaluation criteria given by experts. The higher the risk scores, the more critical the situation is. Table 3.1, Table 3.2 and Table 3.3 summarized the evaluation criteria for severity, occurrence and detect ratings which is normally practice in a semiconductor manufacturing plant, respectively.

Table 3.1: Scale table for severity

Rank	Linguistic terms	Criteria
10	Very high (Liability)	Failure will affect safety or compliance to law
9 ~ 8	High (Reliability/reputation)	Customer impact
7 ~ 6	Moderate (Quality/Convenience)	Major reliability excursions Impacts customer yield Wrong package/par/marking
5 ~ 2	Low (Special handling)	Yield hit, cosmetic
1	None (Unnoticed)	Unnoticed

Table 3.2: Scale table for occurrence

Rank	Linguistic terms	Criteria
10 ~ 9	Very high	Many/shift, many/day
8 ~ 7	High	Many/week, few/week
6 ~ 4	Moderate	Once/week, several/month
3	Low	Once/month
2	Very low	Once/quarter
1	Remote	Once ever

Table 3.3: Scale table for detect

Rank	Linguistic terms	Criteria
10	Extremely low	No control available
9	Very low	Controls probably will not Detect
8 ~ 7	Low	Controls may not detect excursion until reach next functional area
6 ~ 5	Moderate	Controls are able to detect within the same functional area
4 ~ 3	High	Controls are able to detect within the same machine/module
2 ~ 1	Very high	Prevent excursion from occurring

Traditionally, occurrence score is rated and updated manually from time to time by users (based on their knowledge/experience, and with reference to the latest data) in accordance with an occurrence scale table (see Table 3.2). However, it require either a time-consuming process to update the scores manually from time to time or the updating scores are often too subjective based on the user's preferences. Owing to these reason, an FIS-based occurrence model (i.e., an example of a single input fuzzy assessment model) is proposed by (Tay & Lim, 2008). A sampling period (i.e., S) is firstly pre-determined by FMEA users, and the frequency of a failure within S weeks is extracted from a historical database. The occurrence frequency of a failure mode is then averaged to per k_{week} week by using Equation 3.7.

$$\text{Average occurrence of failure mode per k week} = \frac{\text{Number of faliure mode occurrences}}{S} \times k_{week} \tag{3.7}$$

An extended scale table for occurrence is summarized in Table 3.4. Setting $k_{week} = 52$, i.e., one year, is considered. Column "*Occurrence Score*" shows the occurrence scores ranging from 1 to 10. Column "*Linguistic Terms*" shows the assigned linguistic terms. Columns "*Average occurrences of failure modes/52 weeks*" show the lower and upper limits pertaining to each occurrence score, ranging from 1 to 1000. Column "*Criteria*" shows the detailed descriptions of each occurrence score. As an example, $j_1 = 3$ is associated with an occurrence score of 3 and a linguistic term of "*Low*". Its *Average occurrence of failure modes/52 weeks* ranges from 5 to 12, and is explained as "*Once/month*". To ease the modelling process, *Average occurrences of failure modes /52 weeks* is further presented in the logarithm scale, as tabulated in the respective sub-columns of "*Average occurrence of failure modes/52 weeks (log)*". Based on the same example (i.e., $j_1 = 3$), the lower and upper limits are 0.6532 (i.e., log (4.5)) and 1.0969 (i.e., log (12.5)), respectively. The midpoint is obtained by averaging the lower and upper limits.

Table 3.4: An extended scale table for occurrence

j_1	Occurrence Score	Linguistic Terms	Criteria	Average occurrence of failure mode /52 weeks		Average occurrence of failures mode/52 weeks (log)		
				lower limit estimation	upper limit estimation	Lower Limit	Mid-point	Upper Limit
6	10~9	Very High	Many/shift, Many/day	301	1000	2.4778	3.0000	-
5	8~7	High	Many/week, Few/week	53	300	1.7202	2.0990	2.4778
4	6~4	Moderate	Once/week, Several/month, Few/quarter	13	52	1.0969	1.4086	1.7202
3	3	Low	Once/month	5	12	0.6532	0.8751	1.0969
2	2	Very Low	Once/quarter	3	4	0.3979	0.5256	0.6532
1	1	Remote	Once ever	1	2	-	0	0.3979

The FIS-based occurrence model transforms the *Average occurrence of failure modes/52 weeks* into an occurrence score. In other words, the model takes *Average occurrence of failure modes per* k_{week} *week* as the input (denoted as x_1 hereafter) and produces the occurrence score as the output (denoted as y hereafter). The FIS-based occurrence model should satisfy the monotonicity property (see Definitions 3.4.3) between the input and output, i.e., when x_1 increases, y should increase or remain unchanged. Information in Table 3.4 can also be interpreted as a mapping from x_1 to y, and can be represented as fuzzy rules. As an example, for $j_1 = 3$, the expression is as follows.

IF *Average occurrence of failure modes/52 weeks* is "about log (5) to log (12)", **THEN** *Occurrence Score is* "about 3".

Two fuzzy partitions designs, with trapezoidal and triangular fuzzy sets (see Definition 3.2) respectively, illustrated in Figure 3.3 and Figure 3.4 respectively, are

considered. The corresponding scores in the Y domain are represented as a fuzzy singleton, b_{j_1}, i.e., $b_{j_1} = 1,2,3,5,7.5,10$, for $j_1 = 1,2,3,4,5,6$, respectively.

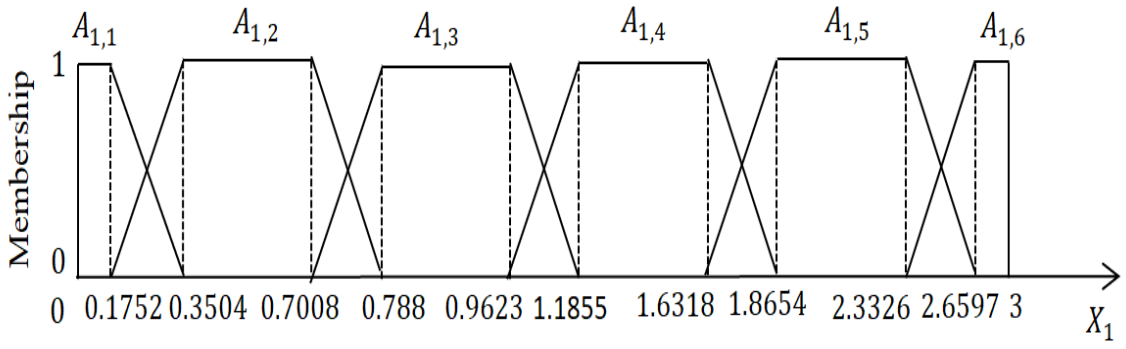

Figure 3.3: Fuzzy partition used for simulation using trapezoidal fuzzy sets

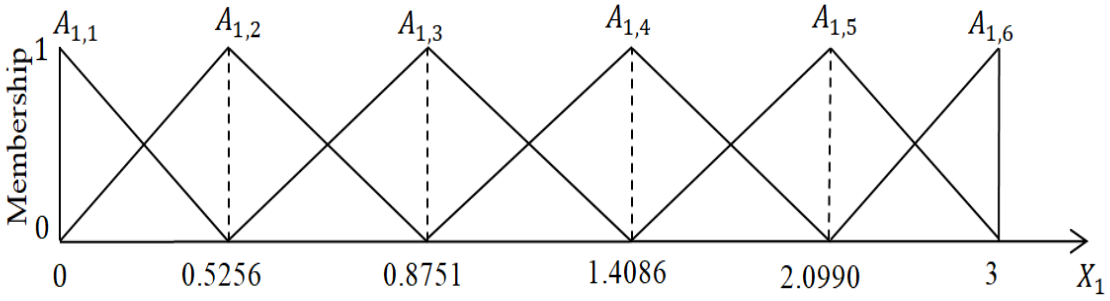

Figure 3.4: Fuzzy partition used for simulation using triangular fuzzy sets

The FIS-based occurrence model can be written as Equation 3.8. The "Occurrence score" versus "Average number of failures occurred/52weeks (log scale)" of the fuzzy occurrence model using trapezoidal and triangular fuzzy set are illustrated in Figure 3.5 and Figure 3.6, respectively. Noted that the "occurrence score" increases monotonically over "Average number of failures occurred/52weeks"; therefore the monotonicity property is fulfilled (DOM $= 1$) using both fuzzy sets.

$$y = f(x_1) = \frac{\sum_{j_1=1}^{k_1} (\mu_{1,j_1}(x_1) \times b_{j_1})}{\sum_{j_1=1}^{k_1} \mu_{1,j_1}(x_1)} \quad (3.8)$$

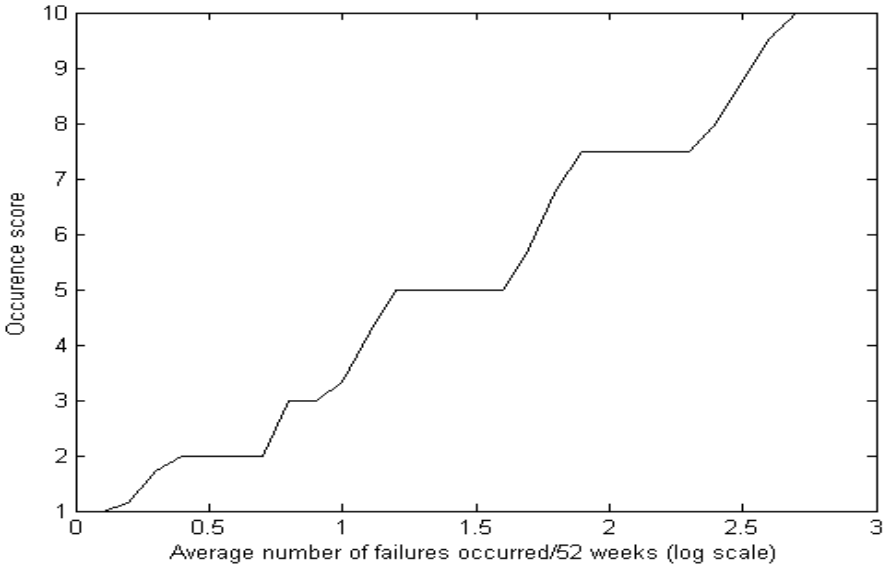

Figure 3.5: Plot of "Occurrence score" versus "Average number of failures occurred/52 weeks (log scale)" using trapezoidal fuzzy set

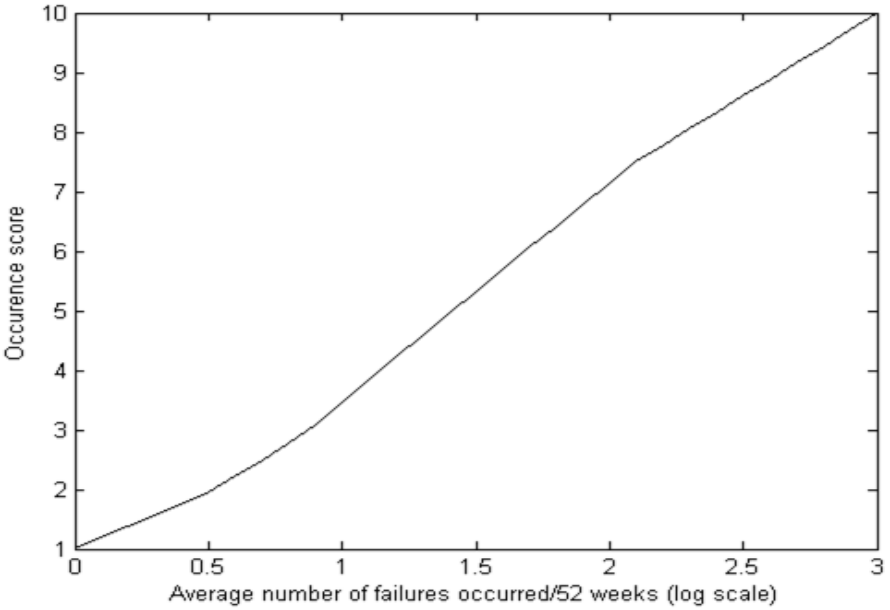

Figure 3.6: Plot of "Occurrence score" versus "Average number of failures occurred/52 weeks (log scale)" using triangular fuzzy set

3.5.2 An FIS-based Risk Model (Markowski & Mannan, 2008)

The fuzzy risk matrix of an FIS-based risk model from the work in (Markowski & Mannan, 2008) is considered. The FIS-based risk model is a two-attribute monotonic function, i.e., severity (Sev) and frequency (in a log scale) (Fre), with respect to the input space of $[1,5] \times [10^{-8}, 10^{0}]$. It produces a risk index, i.e., $risk\ index = f(sev, fre)$, with respect to the output space of $[0,5]$. A proper risk matrix is an important management task especially for the safety policy or safety program, e.g., for SEVESO industry where the risk matrix is included into a major accident prevention policy (Markowski, 2005). The fuzzy partitions for $k_1 = 5, k_2 = 7$ using trapezoidal and triangular fuzzy set are illustrated in Figure 3.7 and Figure 3.8, respectively.

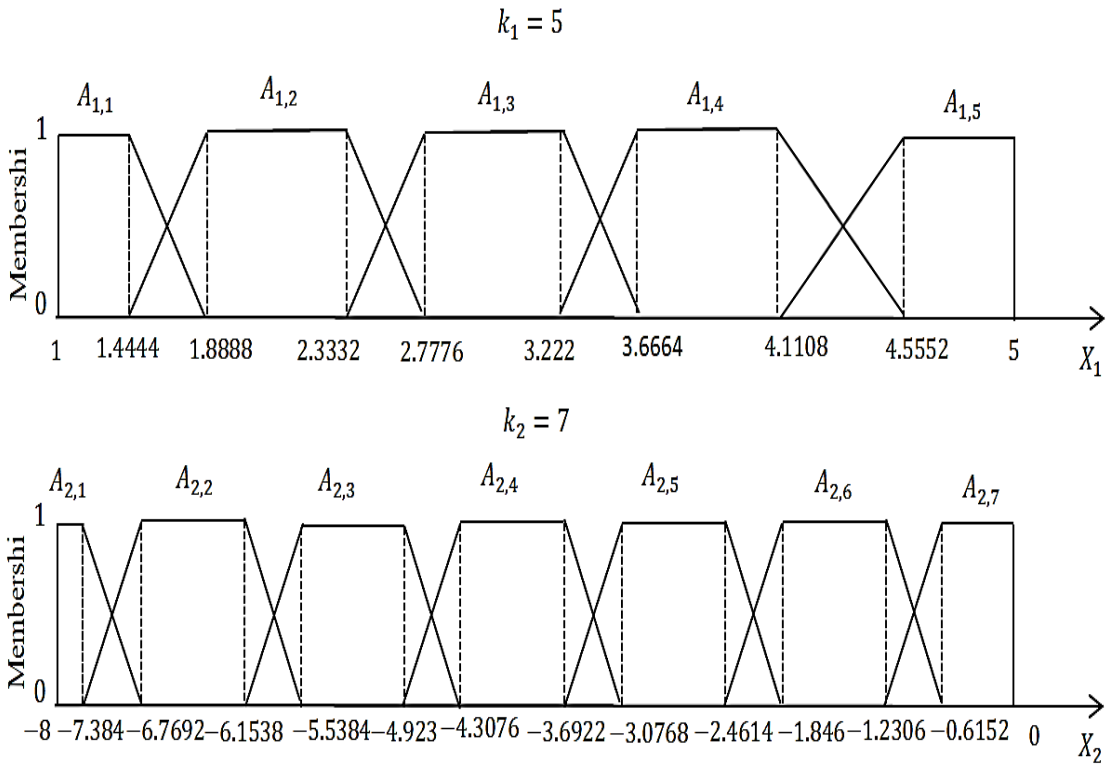

Figure 3.7: Fuzzy partition used for simulation using trapezoidal fuzzy set

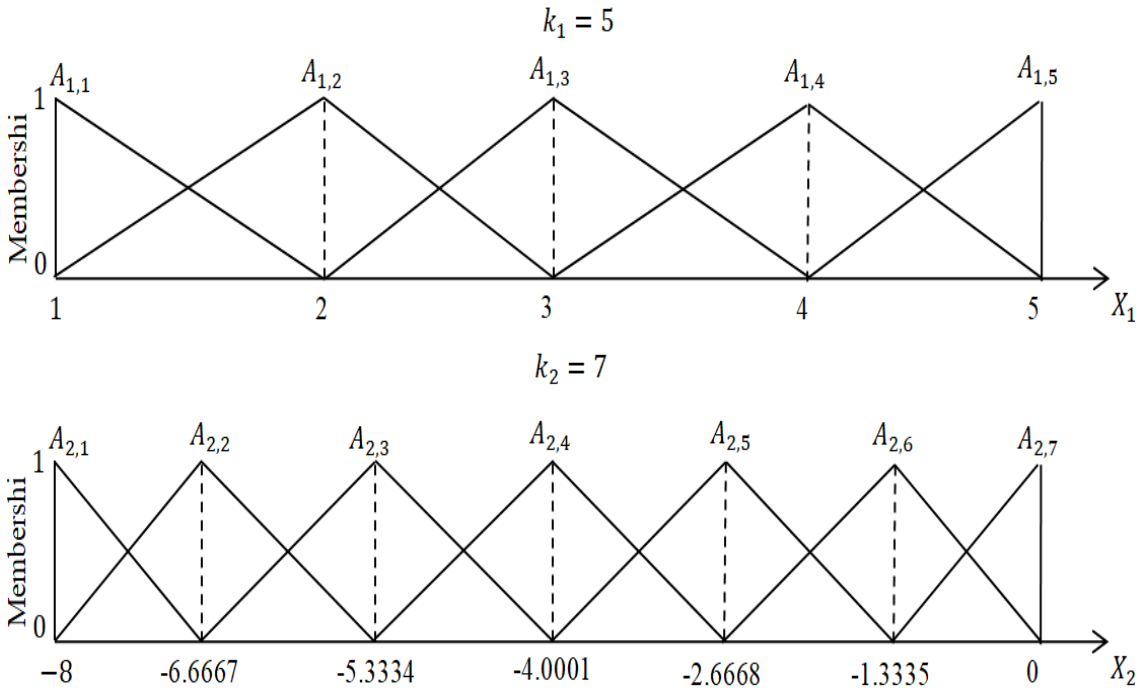

Figure 3.8: Fuzzy partition used for simulation using triangular fuzzy set

The linguistic terms of the fuzzy sets for risk index are categorized into four categories, i.e., (1) A: acceptable (i.e., no further action is required); (2) TA: tolerable acceptable (i.e., further action is based on as low as reasonably practicable (ALARP) principle); (3) TNA: tolerable–unacceptable (i.e., additional safety measures are required); and (4) NA: non-acceptable (i.e., must change immediately). The corresponding risk index for A, TA, TNA and NA are represented as a fuzzy singletons of 1, 2, 3, and 4, respectively. Three fuzzy risk matrixes, i.e., "hard", "standard", and "easy" (Markowski & Mannan, 2008) are illustrated in Table 3.5, Table 3.6, Table 3.7, respectively. For the "hard" fuzzy risk matrix, it requires higher cost, but it is safer. For the "standard" fuzzy risk matrix, it represents a classic fuzzy risk matrix encountered in many industries. For the "easy" fuzzy risk matrix, it offers lower categories of risk where less layer of protection is need for the safety guarantee.

Table 3.5: "Hard" fuzzy risk matrix

$A_{1,5}$	TA	TNA	TNA	NA	NA	NA	NA
$A_{1,4}$	TA	TA	TNA	TNA	NA	NA	NA
$A_{1,3}$	A	TA	TA	TNA	TNA	NA	NA
$A_{1,2}$	A	A	TA	TA	TNA	TNA	NA
$A_{1,1}$	A	A	A	TA	TA	TNA	TNA
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

Table 3.6: "Standard" fuzzy risk matrix

$A_{1,5}$	TA	TA	TNA	TNA	NA	NA	NA
$A_{1,4}$	A	TA	TA	TNA	TNA	NA	NA
$A_{1,3}$	A	A	TA	TA	TNA	TNA	NA
$A_{1,2}$	A	A	A	TA	TA	TNA	TNA
$A_{1,1}$	A	A	A	A	TA	TA	TNA
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

Table 3.7: "Easy" fuzzy risk matrix

$A_{1,5}$	A	TA	TA	TNA	TNA	NA	NA
$A_{1,4}$	A	A	TA	TA	TNA	TNA	NA
$A_{1,3}$	A	A	A	TA	TA	TNA	TNA
$A_{1,2}$	A	A	A	A	TA	TA	TNA
$A_{1,1}$	A	A	A	A	A	TA	TA
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

The surface plots of risk index with respect to severity and the frequency (in a log scale) for the FIS model using trapezoidal and triangular fuzzy sets for "hard" fuzzy risk matrix are illustrated in Figure 3.9 and Figure 3.10, respectively. *Monotone* surface curve (i.e.,$\mathrm{DOM} = 1$) can be observed for both trapezoidal and triangular fuzzy sets.

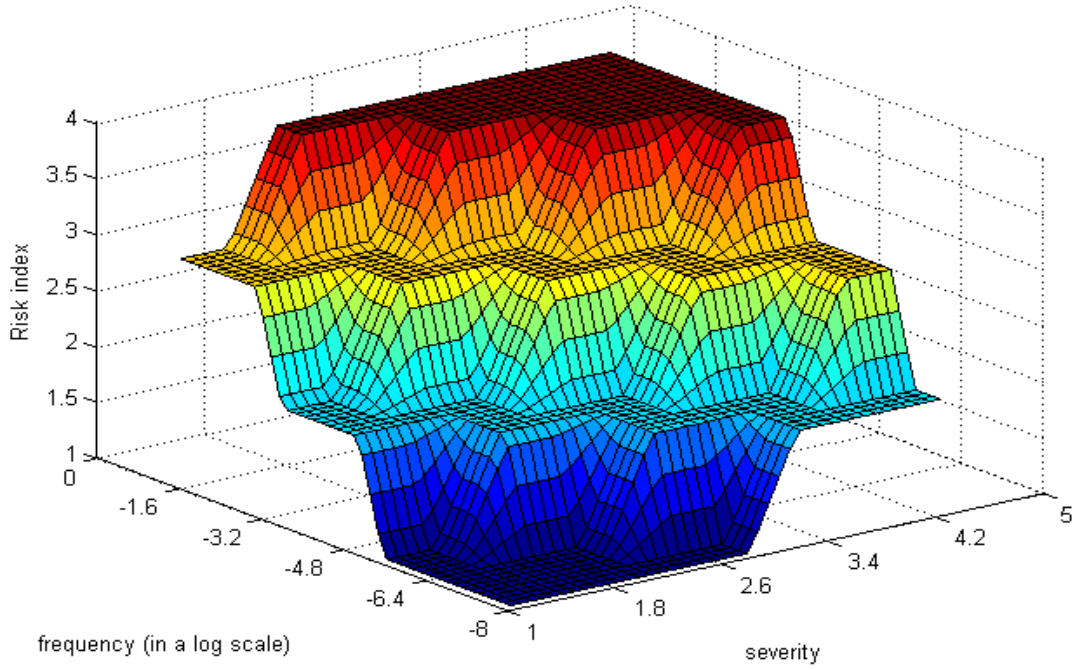

Figure 3.9: Surface plots for the FIS model using trapezoidal fuzzy set for "hard" fuzzy risk matrix

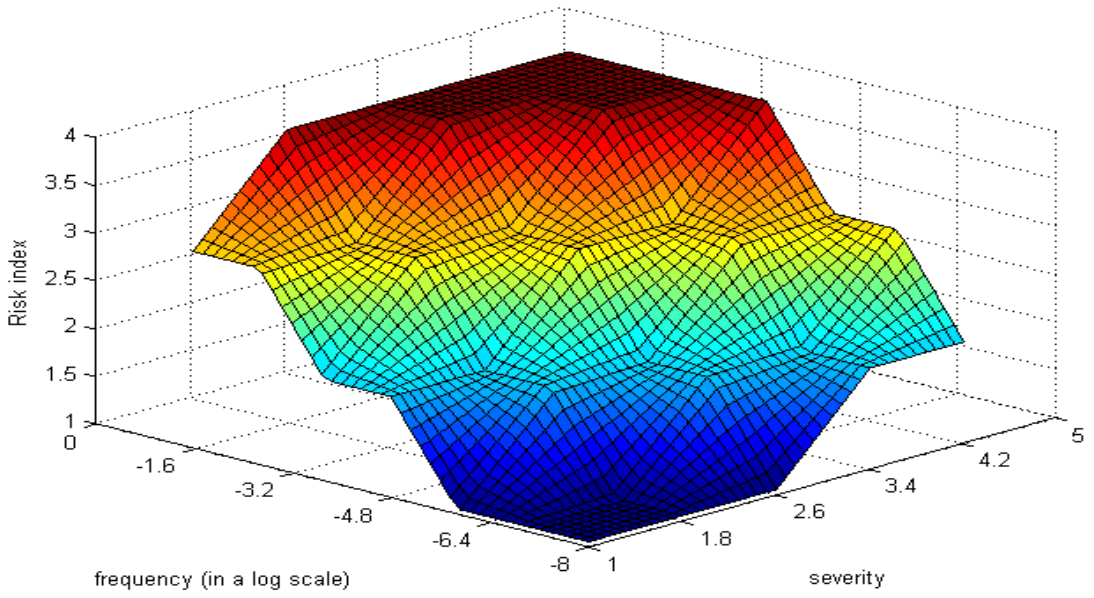

Figure 3.10: Surface plots for the FIS model using triangular fuzzy set for "hard" fuzzy risk matrix

The surface plots of risk index with respect to severity and the frequency (in a log scale) for the FIS model using trapezoidal and triangular fuzzy sets for "standard" fuzzy risk matrix are illustrated in Figure 3.11 and Figure 3.12, respectively. Again, *monotone* surface curve (i.e.,DOM = 1) can be observed for both trapezoidal and triangular fuzzy sets.

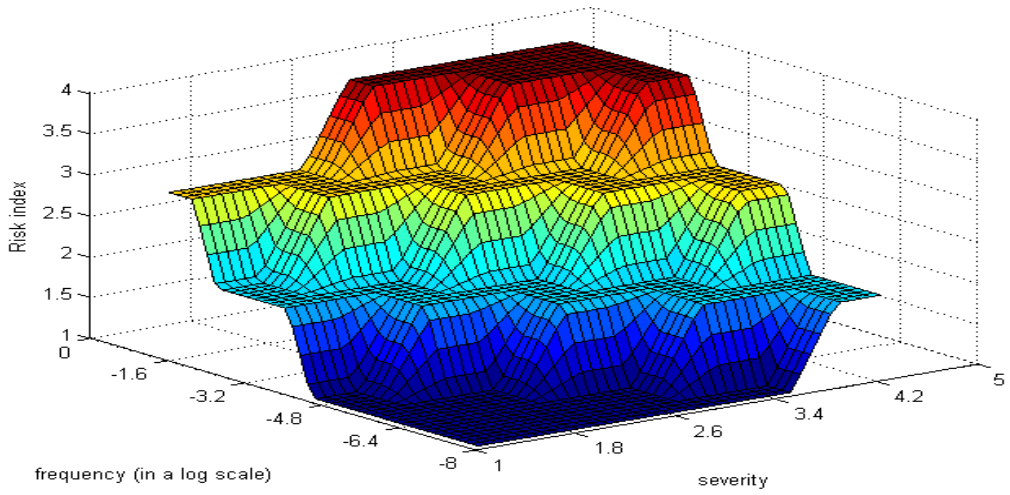

Figure 3.11: Surface plots for the FIS model using trapezoidal fuzzy set for "standard" fuzzy risk matrix

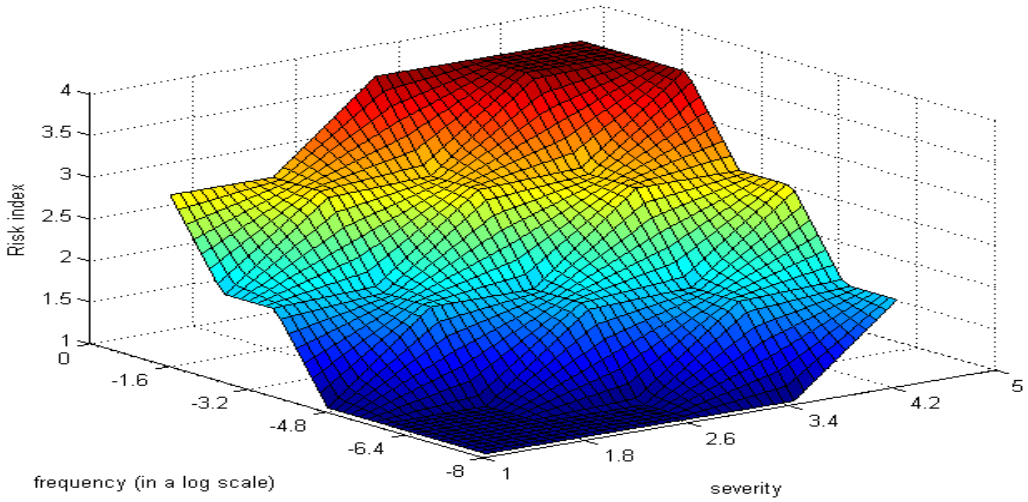

Figure 3.12: Surface plots for the FIS model using triangular fuzzy set for "standard" fuzzy risk matrix

The surface plots of risk index with respect to severity and the frequency (in a log scale) for the FIS model using trapezoidal and triangular fuzzy sets for "easy" fuzzy risk matrix are illustrated in Figure 3.13 and Figure 3.14, respectively. And again, *Monotone* surface curve (i.e.,DOM = 1) can be observed for both trapezoidal and triangular fuzzy sets.

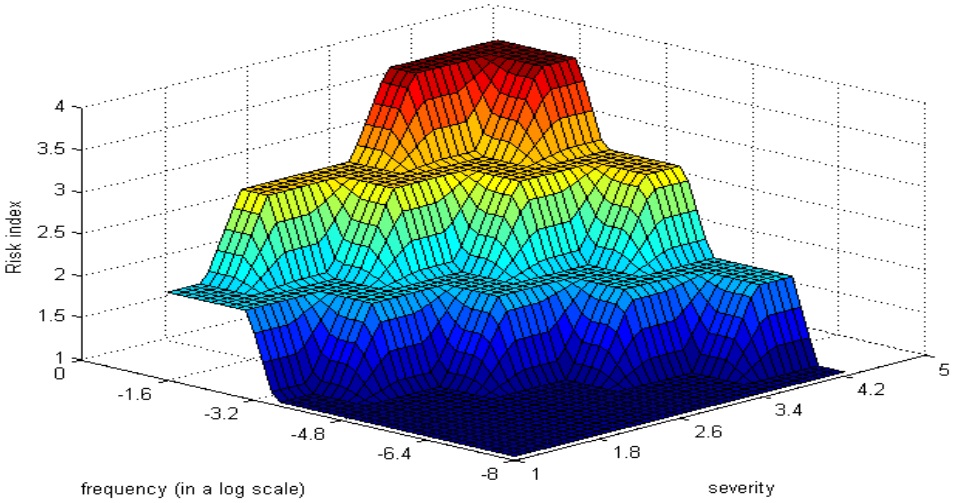

Figure 3.13: Surface plots for the FIS model using trapezoidal fuzzy set for "easy" fuzzy risk matrix

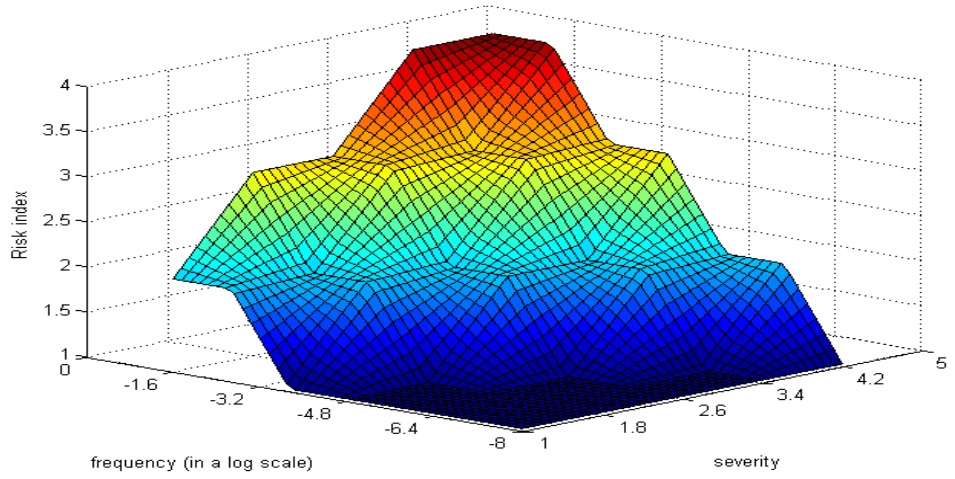

Figure 3.14: Surface plots for the FIS model using triangular fuzzy set for "easy" fuzzy risk matrix

3.6 Summary

In this chapter, an analysis of the zero-order TSK FIS model is presented. We show that by using a set of *sufficient conditions* (i.e., convex and normal trapezoidal/triangular membership function coupled with a strong fuzzy partition strategy), a zero-order TSK FIS model always satisfy the monotonicity property if the fuzzy rules are *monotone* and *complete* (see Definition 3.4) A monotone test is established to evaluate the monotonicity property of an FIS model. Two benchmark examples are presented. The usefulness of these proposals is illustrated with benchmark information, with positive results obtained.

In the next chapter, *monotone data sets* are defined and a number of simulated examples are introduced. Analyses of the *ad hoc* methods for single and multi-attribute *monotone* data sets are presented. A six-stage framework for practical modeling of *monotone* TSK FIS models with *ad hoc* methods is proposed. Experiment results are conducted to demonstrate the usefulness of our proposed approach.

CHAPTER 4

AD HOC METHODS WITH MONOTONE DATA

4.1 Chapter Overview

In the previous chapter, a set of *sufficient conditions* for the zero-order TSK FIS model was presented. Besides that, a *monotone* test procedure to evaluate the monotonicity property of an FIS model was devised. In this chapter, *monotone data set* (see Definition 4.1) is considered. An analysis for two *ad hoc* methods (i.e., WM and Ishibuchi methods, Equations 4.3 and 4.4, respectively) are conducted to ascertain whether both WM and Ishibuchi methods are able to always produce *monotone* fuzzy rules with single and multi-attribute *monotone* data sets. However, our analytical results indicate that fuzzy If-Then rules generated using both *ad hoc* methods: (1) always satisfy the monotonicity property when a single-attribute *monotone* data set is considered; (2) do not always satisfy the monotonicity property when a multi-attribute *monotone* data set is considered. Therefore, a six-stage framework for practical modeling of *monotone* TSK FIS models with *ad hoc* methods is proposed. All the experiment has been carried out using MATLAB ® R2012a and a number of simulated examples are used to demonstrate the usefulness of our proposed framework for modeling monotone TSK FIS models with *ad hoc* methods.

In Section 4.2, the *monotone* data sets are defined and a number of simulated examples are introduced. In Section 4.3, analyses of the *ad hoc* methods for single and multi-attribute *monotone* data sets are presented. In Section 4.4, a six-stage framework for practical modeling of *monotone* TSK FIS models with *ad hoc* methods is proposed. Finally, concluding remarks are given in Section 4.5.

4.2 Monotone Data

Definition 4.1: Consider a set of input-output data sample, i.e., $(\boldsymbol{x}^p; y^p), p \in [1,2,\cdots,h]$, where output $y^p \in Y$ is influenced by n-input variables, $\boldsymbol{x}^p = [x_1^p, x_2^p, \cdots, x_n^p] \in X_1 \times X_2 \times \cdots \times X_n$ based on the data set. A data sample, y^q, is further considered, where $q \in [1,2,\cdots,h]$ and $q \neq p$.

4.1.1 Two notions are defined, as follows.

$$\underline{y^q} = \max\{y^p | \text{ all samples } p \neq q \text{ for which } x_i^p \leq x_i^q, i = 1,\cdots,n\} \quad (4.1)$$

$$\overline{y^q} = \min\{y^p | \text{ all samples } p \neq q \text{ for which } x_i^p \geq x_i^q, i = 1,\cdots,n\} \quad (4.2)$$

4.1.2 The data set is *monotone* if $\underline{y^q} \leq y^q \leq \overline{y^q}$ is true for all $q = 1,\cdots,h$. Otherwise, the data set is *non-monotone*.

4.1.3 The data set is *complete* if all $[x_1^p, x_2^p, \cdots, x_n^p; y^p]$ is known for all $p = 1,\cdots,h$.

A monotone data set can be obtained by pre-processing a non-monotone data set, either by removing the *non-monotone* data samples (Rademaker, De Baets, & De Meyer, 2009) or using a *monotone* data relabeling procedure (H Daniels & Velikova, 2003).

4.2.1 Simulated Example 4.1: Single Attribute *Monotone* Data

Example 4.1: Consider a single attribute *monotone* data set with 101 input-output pairs, i.e., $(x_1^p; y^p), p = 1,2,\cdots,101$. The data set is generated using a single attribute monotonic

function, i.e., $y = 0.5tanh(x_1)$ with respect to the input space [0,1], as shown in Figure 4.1.

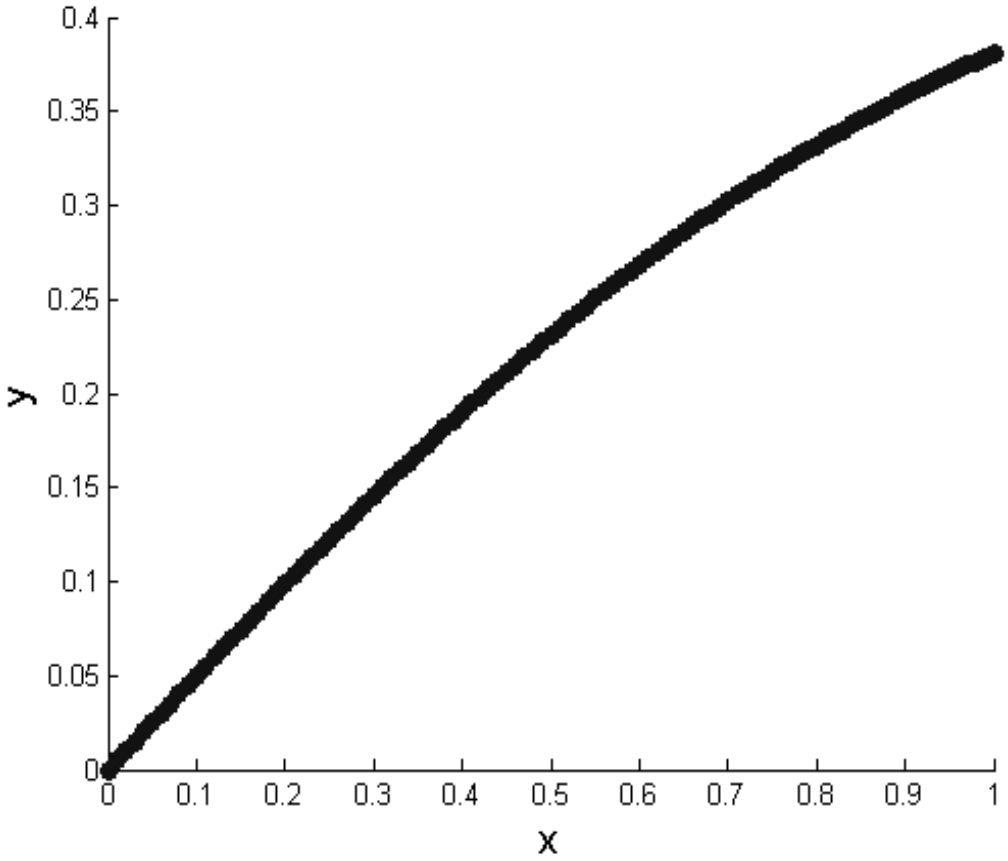

Figure 4.1: *Monotone* data sets for Example 4.1

The fuzzy partitions for $k_1 = 3$ using trapezoidal and triangular fuzzy set are illustrated in Figure 4.2 and Figure 4.3, respectively. The fuzzy partitions for $k_1 = 5$ using trapezoidal and triangular fuzzy set are illustrated in Figure 4.4 and Figure 4.5, respectively. The fuzzy partitions for $k_1 = 7$ using trapezoidal and triangular fuzzy set are illustrated in Figure 4.6 and Figure 4.7, respectively.

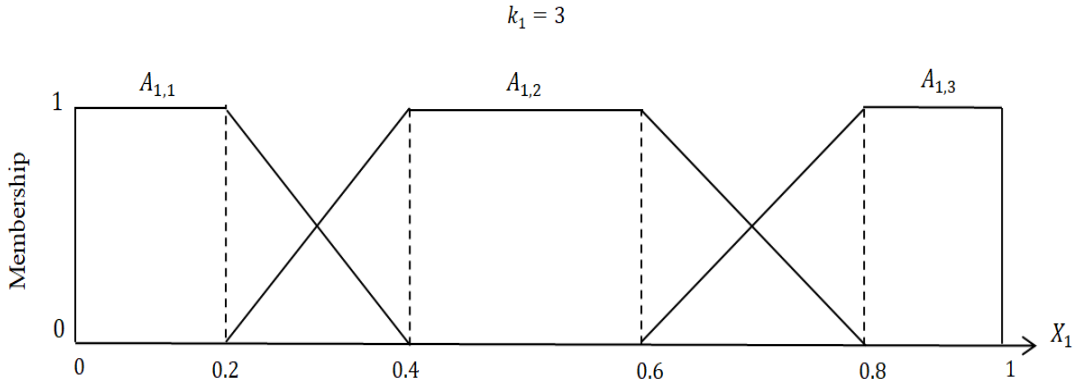

Figure 4.2: Fuzzy partition for $k_1 = 3$ using trapezoidal fuzzy set

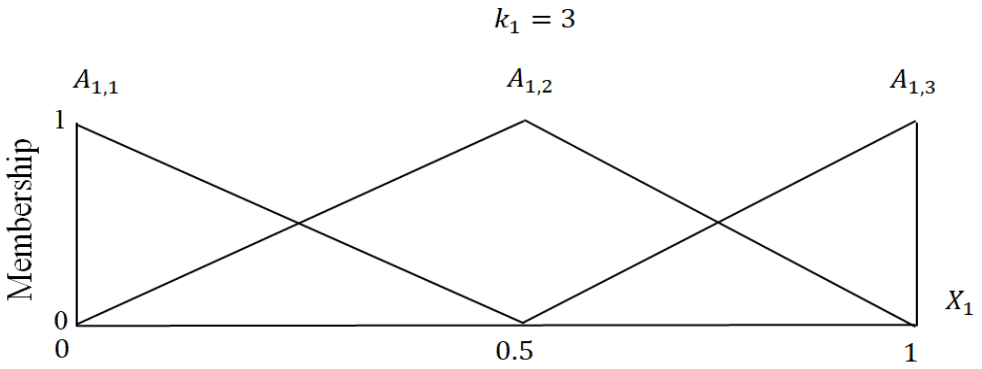

Figure 4.3: Fuzzy partition for $k_1 = 3$ using triangular fuzzy set

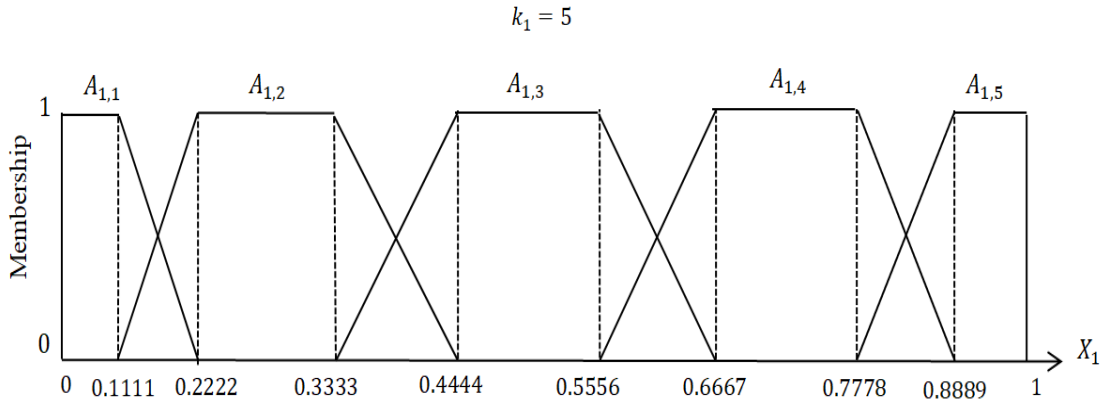

Figure 4.4: Fuzzy partition for $k_1 = 5$ using trapezoidal fuzzy set

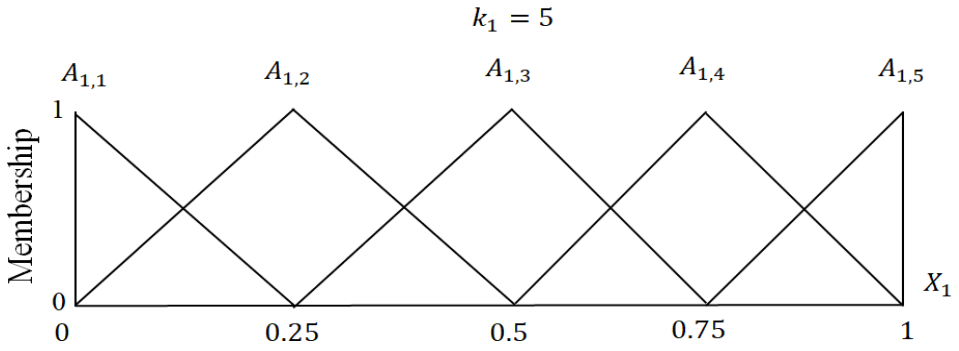

Figure 4.5: Fuzzy partition for $k_1 = 5$ using triangular fuzzy set

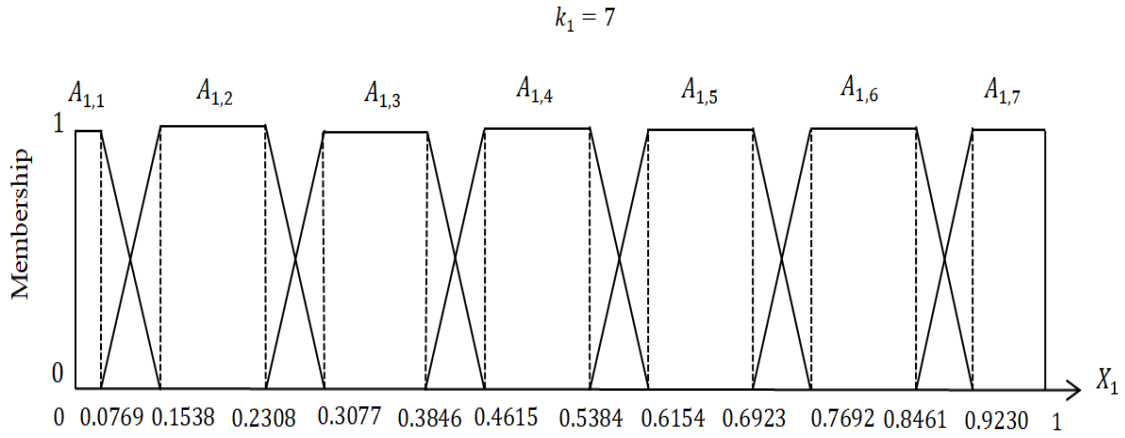

Figure 4.6: Fuzzy partition for $k_1 = 7$ using trapezoidal fuzzy set

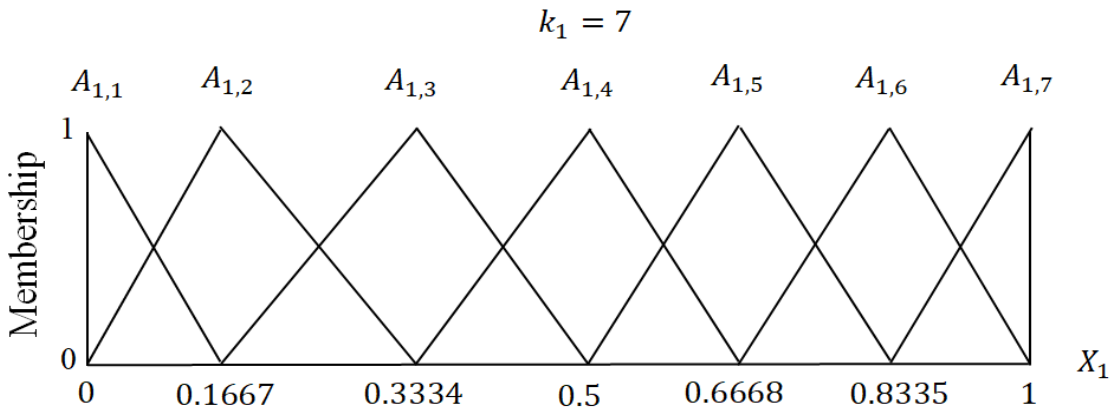

Figure 4.7: Fuzzy partition for $k_1 = 7$ using triangular fuzzy set

4.2.2 Simulated Example 4.2: Multi-attribute *Monotone* Data

Example 4.2: A two-attribute *monotone* data set with 121 input-output pairs, i.e., $(\boldsymbol{x}^p; y^p), p = 1,2,\cdots,121$, is considered. Note that $\boldsymbol{x}^p = \left(x_1^p, x_2^p\right)$ and y^p. The data set is generated using a multi-attribute monotonic function, i.e., $y = \frac{1}{4}[2 + tanh(0.4x_1) + tanh(0.4x_2)]$ with respect to the input space of $[-5,5] \times [-5,5]$. A total of 74 input-output pairs are chosen as the training set (i.e.,$h = 74$), as depicted in Figure 4.8. Another 47 input-output pairs are chosen as the validation set, as depicted in Figure 4.9.

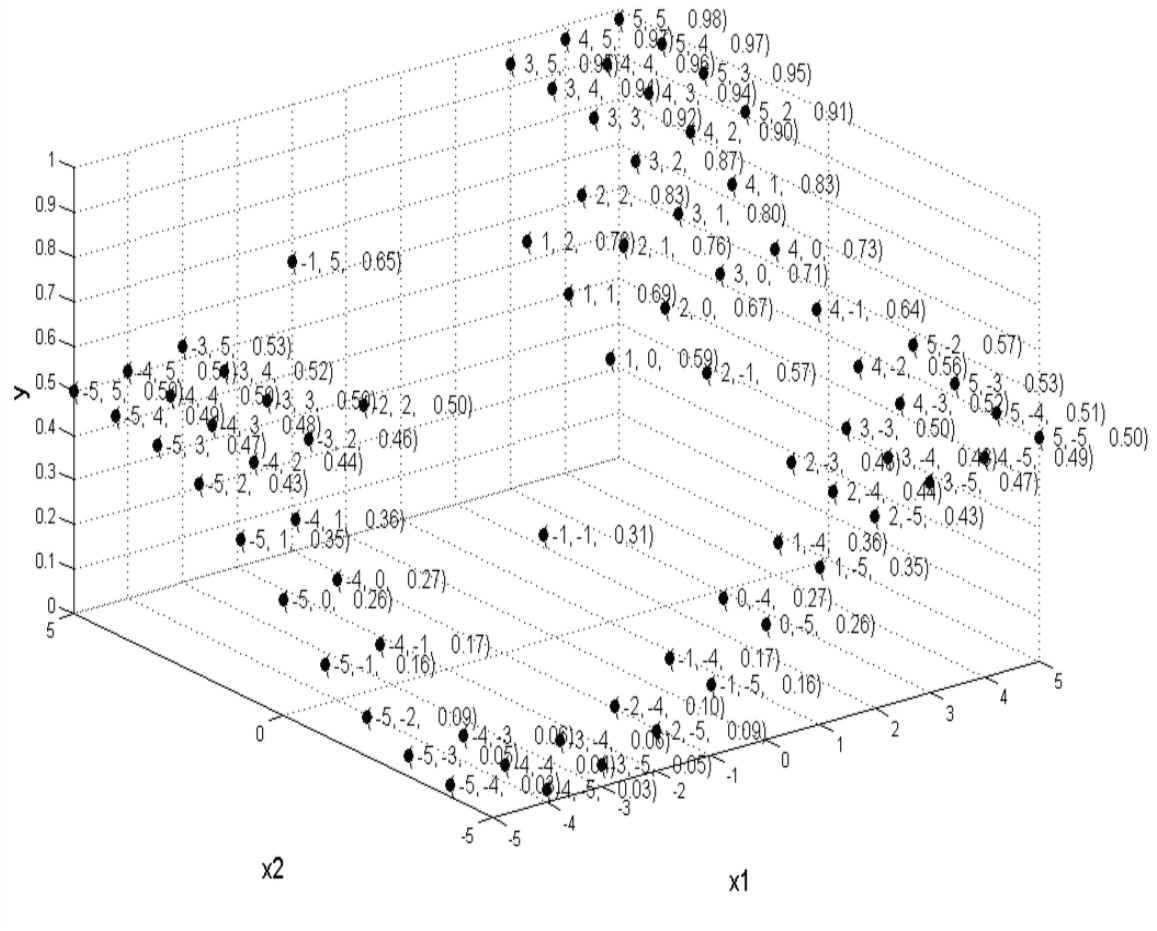

Figure 4.8: *Monotone* training data for Example 4.2

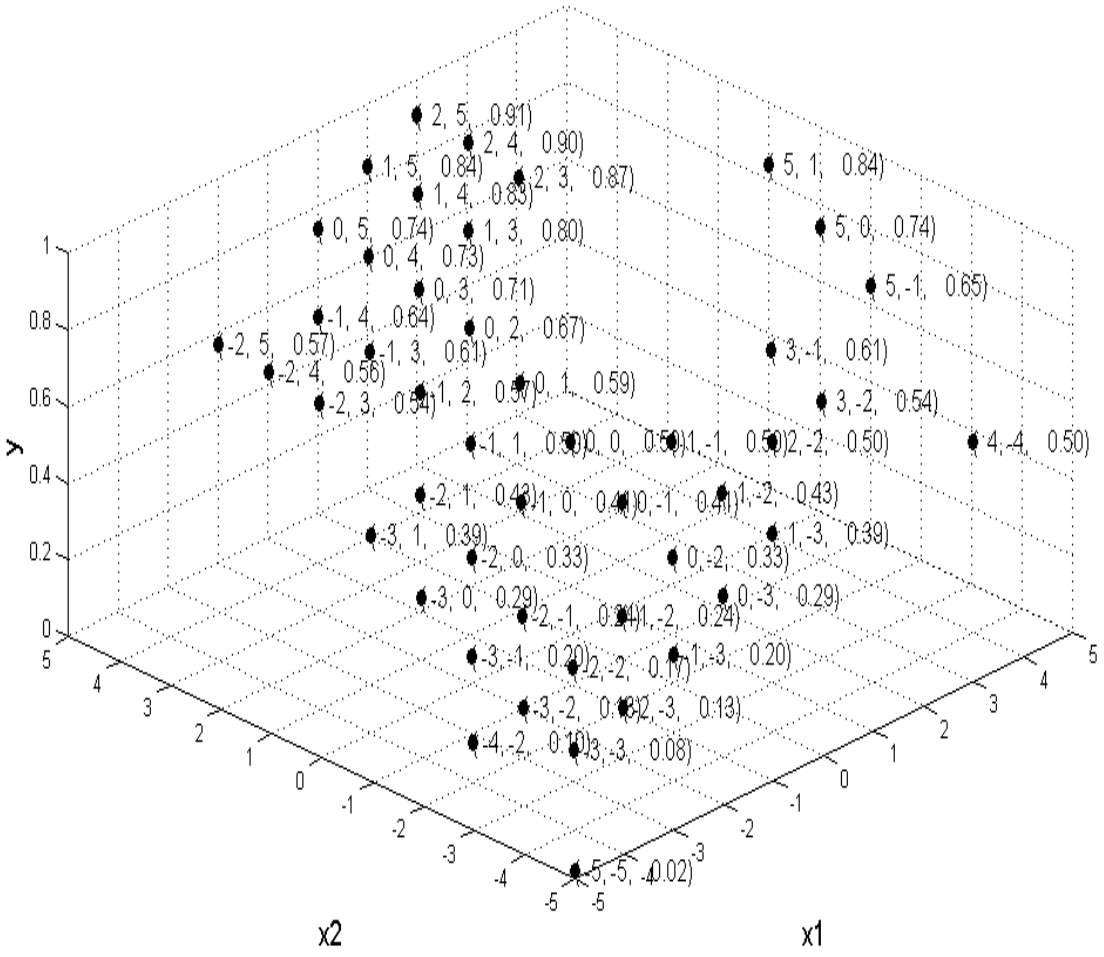

Figure 4.9: *Monotone* validation data for Example 4.2

The fuzzy partitions for $k_1 = k_2 = 3$ using trapezoidal and triangular fuzzy set are illustrated in Figure 4.10 and Figure 4.11, respectively. The fuzzy partitions for $k_1 = k_2 = 5$ using trapezoidal and triangular fuzzy set are illustrated in Figure 4.12 and Figure 4.13, respectively. The fuzzy partitions for $k_1 = k_2 = 7$ using trapezoidal and triangular fuzzy set are illustrated in Figure 4.14 and Figure 4.15, respectively.

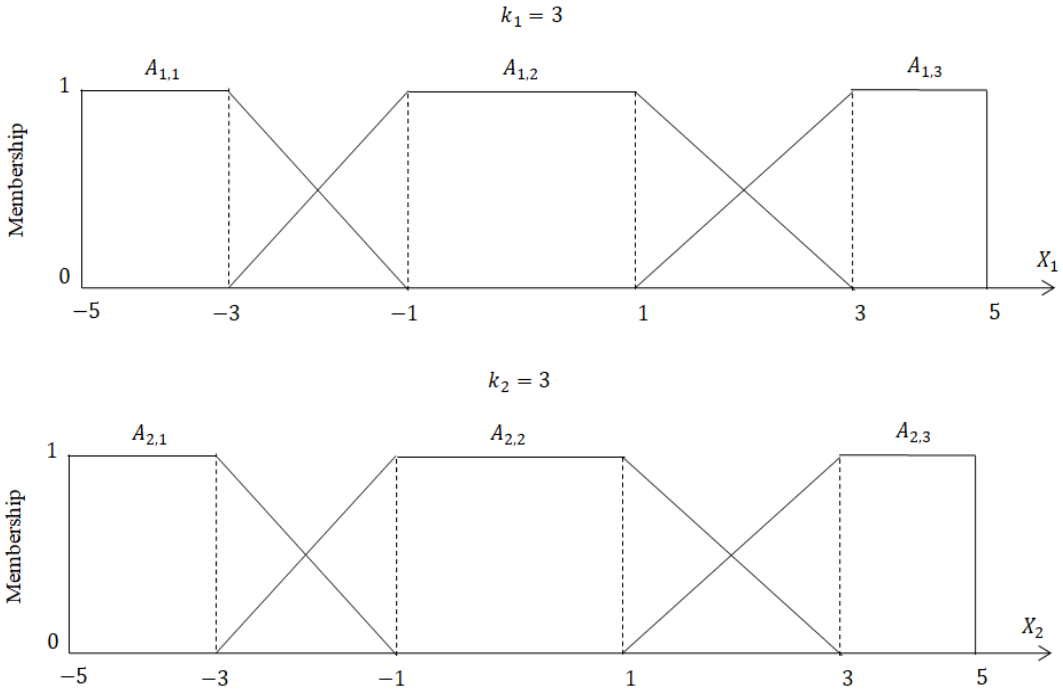

Figure 4.10: Fuzzy partition for $k_1 = k_2 = 3$ using trapezoidal fuzzy set

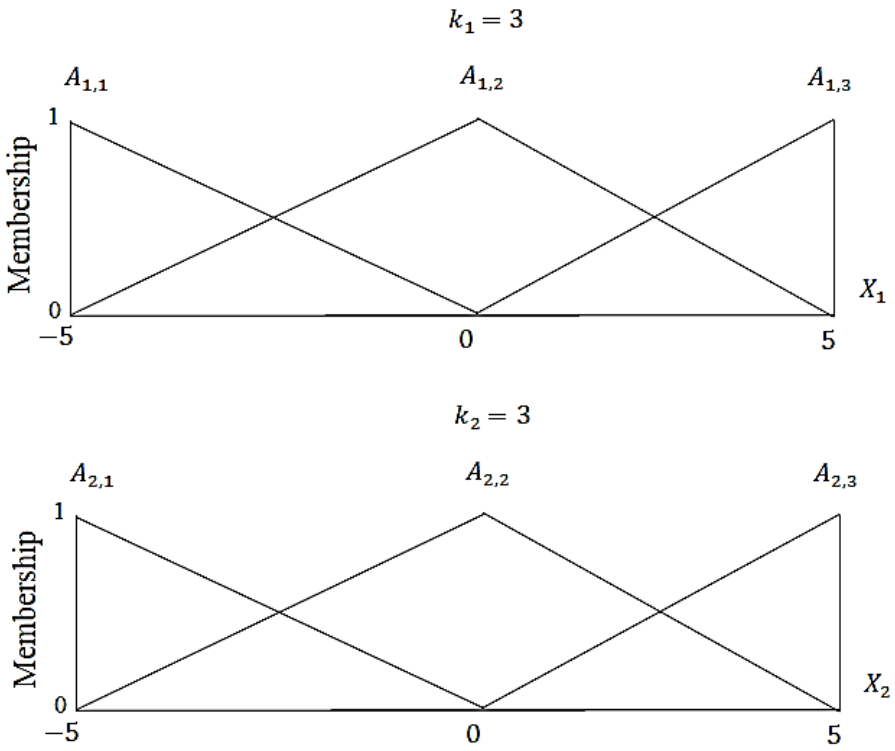

Figure 4.11: Fuzzy partition for $k_1 = k_2 = 3$ using triangular fuzzy set

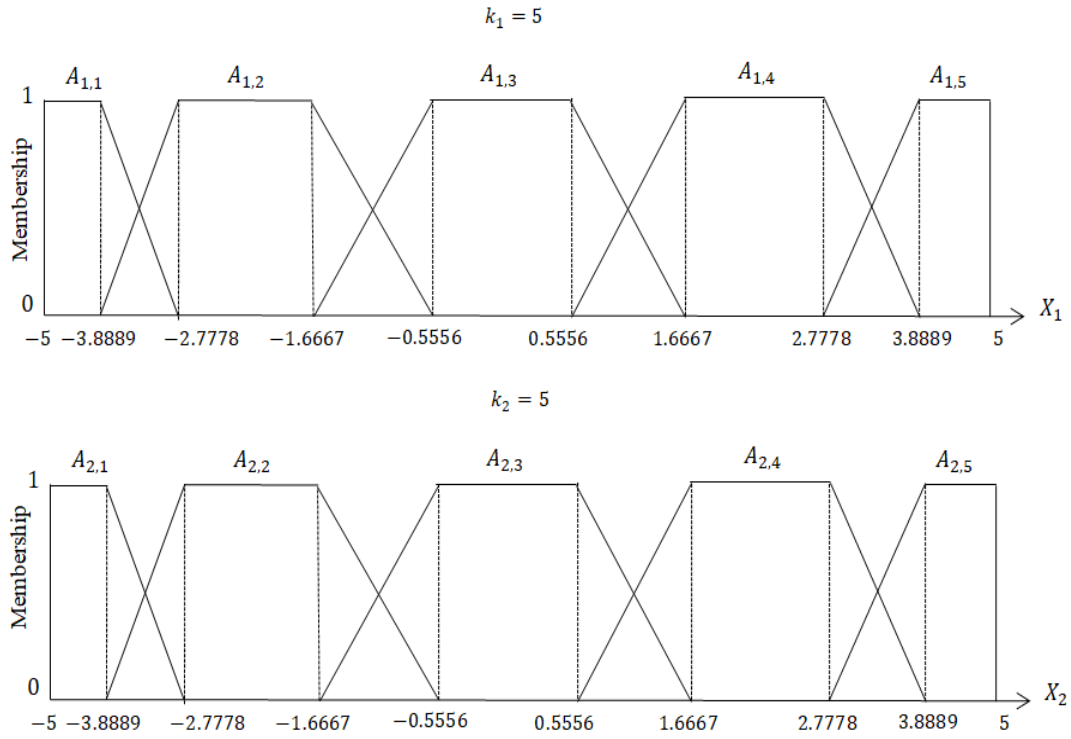

Figure 4.12: Fuzzy partition for $k_1 = k_2 = 5$ using trapezoidal fuzzy set

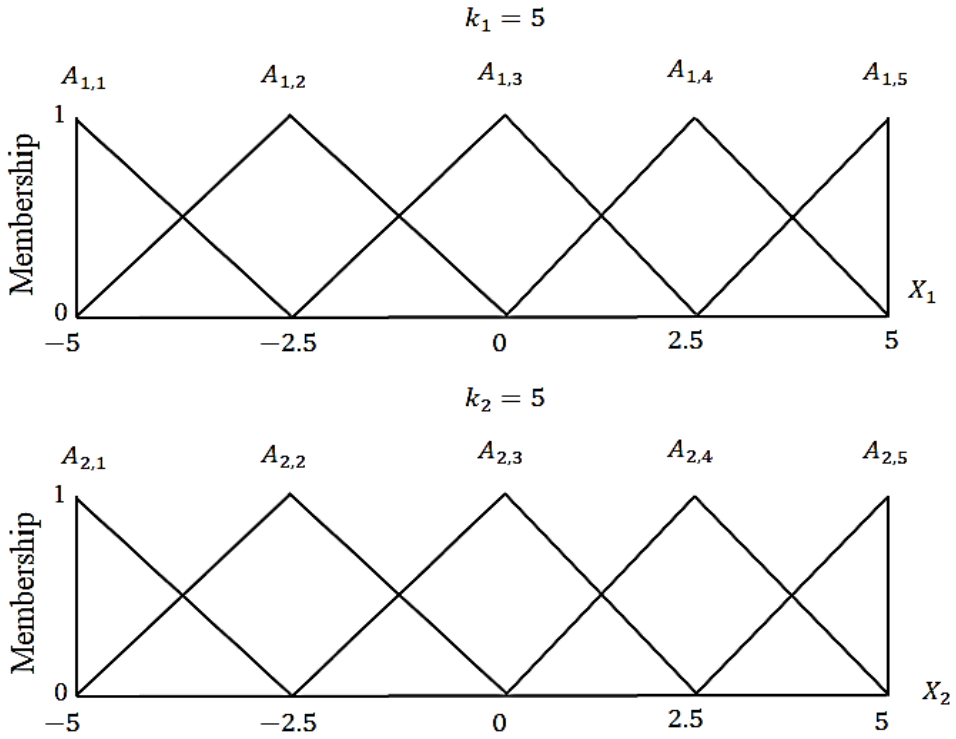

Figure 4.13: Fuzzy partition for $k_1 = k_2 = 5$ using triangular fuzzy set

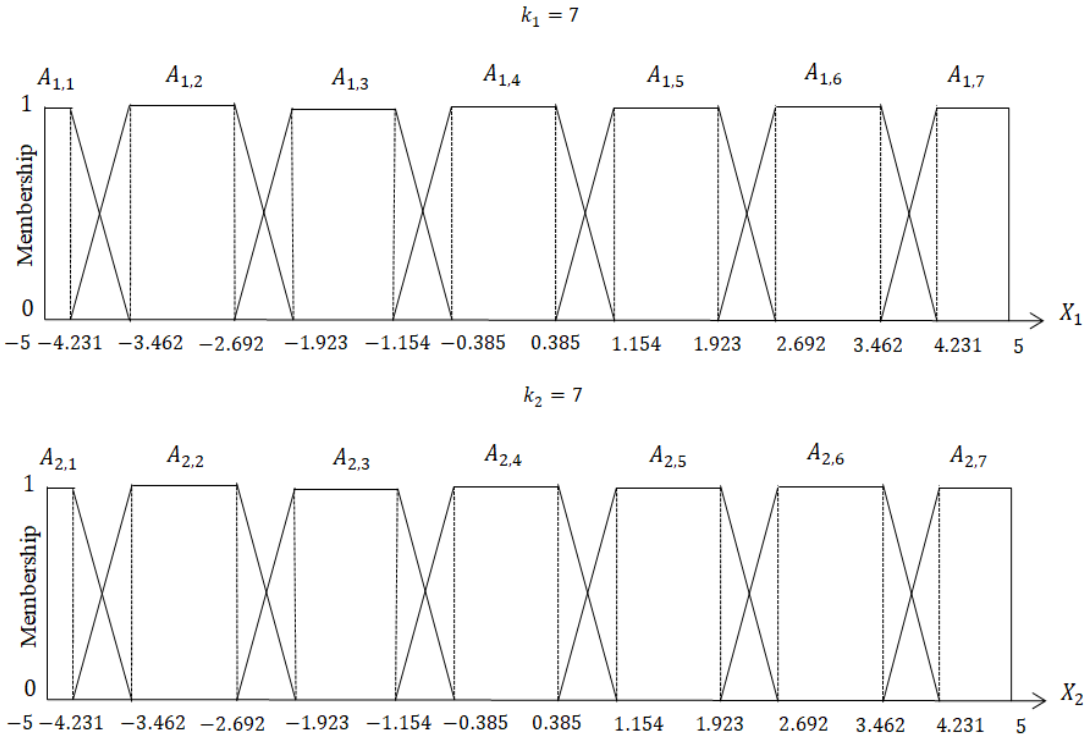

Figure 4.14: Fuzzy partition for $k_1 = k_2 = 7$ using trapezoidal fuzzy set

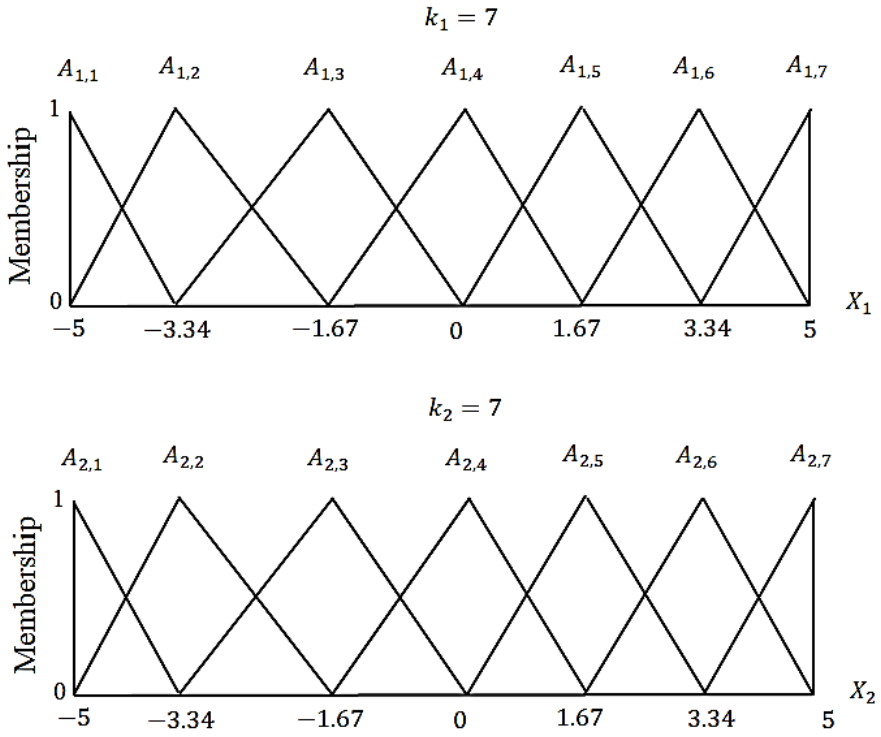

Figure 4.15: Fuzzy partition for $k_1 = k_2 = 7$ using triangular fuzzy set

4.3 *Ad Hoc* Methods

In this section, two different methods to produce data-driven FIS models with *ad hoc* methods (Nozaki et al., 1997), i.e., (1) Wang et al. (Wang, 2003; Wang & Mendel, 1992) (hereafter abbreviated as the Wang-Mendel (WM) method); (2) Ishibuchi et al. (Hisao Ishibuchi et al., 1994) (hereafter abbreviated as the Ishibuchi method) using single and multi-attribute *monotone* data are analyzed. The classification terminologies in (Jorge Casillas et al., 2002) is adopted, in which both WM (Wang, 2003; Wang & Mendel, 1992) and Ishibuchi (Ishibuchi et al., 1994; Nozaki et al., 1997) methods are categorized as "*ad hoc* methods".

Definition 4.2: The *set of dominant rules* for data point $\boldsymbol{x}^p$**,** denoted R^p, is a collection of all fuzzy rules $R_{\boldsymbol{j}}$ having the highest *activation degree*, i.e., $\mu_{\boldsymbol{j}}(\boldsymbol{x}^p) = \prod_{i=1}^{n} \mu_{i,j_i}\left(x_i^p\right) \geq \mu_{\boldsymbol{j}'}(\boldsymbol{x}^p)$ for all integer vectors $\boldsymbol{j}' \neq \boldsymbol{j}$**.**

Using the WM method or Ishibuchi method, $b_{\boldsymbol{j}}$ can be obtained using Equation 4.3 (Wang, 2003; Wang & Mendel, 1992) or Equation 4.4 (Ishibuchi et al., 1994; Nozaki et al., 1997), respectively. If $\mu_{\boldsymbol{j}}(\boldsymbol{x}^p) = 0$ for all integer vectors $\boldsymbol{j}$**,** then R^p is an empty set, and we set $b_{\boldsymbol{j}} = 'unknown'$ for both methods.

$$b_{\boldsymbol{j}}^{\mathrm{WM}} = \frac{\sum_{\forall p: R_{\boldsymbol{j}} \in R^p} \mu_{\boldsymbol{j}}(\boldsymbol{x}^p) y^p}{\sum_{\forall p: R_{\boldsymbol{j}} \in R^p} \mu_{\boldsymbol{j}}(\boldsymbol{x}^p)} \tag{4.3}$$

$$b_{\boldsymbol{j}}^{\mathrm{Ishibuchi}} = \frac{\sum_{\forall p} \mu_{\boldsymbol{j}}(\boldsymbol{x}^p) y^p}{\sum_{\forall p} \mu_{\boldsymbol{j}}(\boldsymbol{x}^p)} \tag{4.4}$$

4.3.1 Analysis using Single Attribute *Monotone* Data

Remark 4.1: An analysis of the WM method (Wang, 2003; Wang & Mendel, 1992) for a *monotone* single attribute input data set, $n = 1$, is summarized, as follows.

4.1.1 The lower limit of b_{j_1}, denoted $\underline{b_{j_1}}$, can be obtained with $\underline{b_{j_1}} = min_{\forall p:R_{j_1} \in R^p}(y^p)$, for $j_1 = 1,2,\cdots,k_1$.

4.1.2 The upper limit of b_{j_1}, denoted $\overline{b_{j_1}}$, can be obtained with $\overline{b_{j_1}} = max_{\forall p:R_{j_1} \in R^p}(y^p)$, for $j_1 = 1,2,\cdots,k_1$

Proposition 4.1: Consider a *monotone* single attribute input data set. By using Equation 4.3, the resulting fuzzy rules are always *monotone* (see Definition 3.4.3).

Proof. We consider any pair of consecutive fuzzy rules, b_{j_1} and b_{j_1+1} for $j_1 = 1,2,\cdots,k_1 - 1$ using Equation 4.3. The upper limit of b_{j_1} can be obtained with $\overline{b_{j_1}} = max_{\forall p:R_{j_1} \in R^p}(y^p)$. The lower limit of b_{j_1+1} can be obtained with $\underline{b_{j_1+1}} = min_{\forall p:R_{j_1+1} \in R^p}(y^p)$. Suppose that the data set is *monotone*, then $\underline{b_{j_1+1}} \geq \overline{b_{j_1}}$ is always true. Therefore, by using Equation 4.3, the resulting fuzzy rules are always *monotone* for a single attribute *monotone* input data set.

Remark 4.2: An analysis of the Ishibuchi method (Ishibuchi et al., 1994; Nozaki et al., 1997) for a single attribute *monotone* input data set, $n = 1$, is summarized, as follows.

Three regions (i.e., Regions A, B and C) as shown in Figure 4.16 are considered. Region (A) covers interval $\left[c_{1,j_1-1} = a_{1,j_1}, c_{1,j_1} = a_{1,j_1+1}\right]$ of the input domain. Data samples fall in Region (A) are denoted $x_{1,A}^p \in \left[c_{1,j_1-1} = a_{1,j_1}, c_{1,j_1} = a_{1,j_1+1}\right] \rightarrow y_A^p \in$

$\left[\underline{y_A^p}, \overline{y_A^p}\right]$. Region (B) covers interval $\left[c_{1,j_1} = a_{1,j_1+1}, d_{1,j_1} = b_{1,j_1+1}\right]$ of the input domain. Data samples fall in Region (B) are denoted $x_{1,B}^p \in \left[c_{1,j_1} = a_{1,j_1+1}, d_{1,j_1} = b_{1,j_1+1}\right] \rightarrow y_B^p \in \left[\underline{y_B^p}, \overline{y_B^p}\right]$. Region (C) covers interval $\left[d_{1,j_1} = b_{1,j_1+1}, d_{1,j_1+1} = b_{1,j_1+2}\right]$ of the input domain. Data samples fall in Region (C) are denoted $x_{1,C}^p \in \left[d_{1,j_1} = b_{1,j_1+1}, d_{1,j_1+1} = b_{1,j_1+2}\right] \rightarrow y_C^p \in \left[\underline{y_C^p}, \overline{y_C^p}\right]$. For a *monotone* data set, $y_A^p \leq y_B^p \leq y_C^p$ is always true.

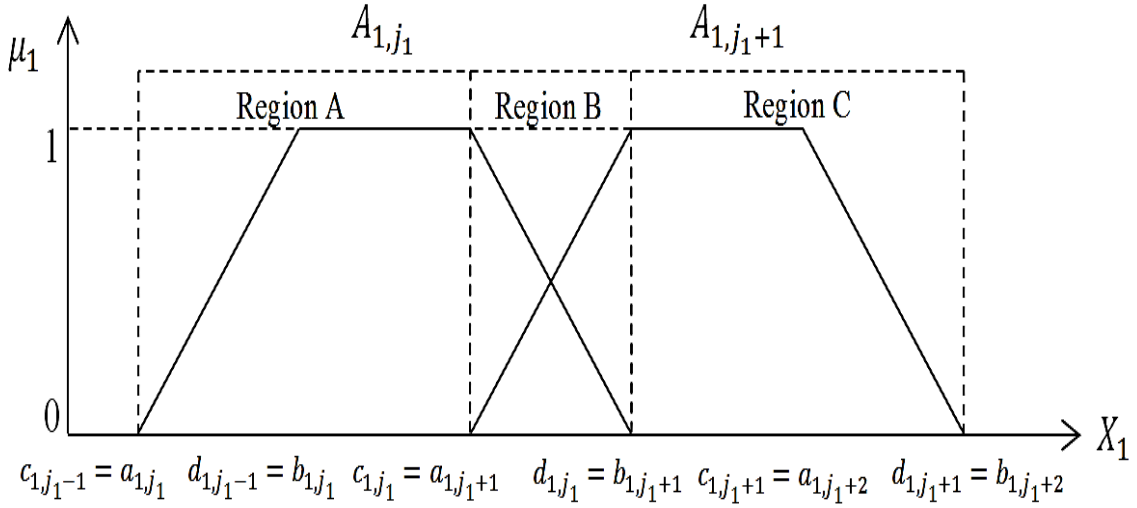

Figure 4.16: Two trapezoidal fuzzy sets with Regions A, B, and C

4.2.1 Re-organizing Equation 4.4, we have

$$b_{j_1} = \frac{\sum_{\forall x_{1,A}^p} \mu_{1,j_1,A}\left(x_{1,A}^p\right) y_A^p + \sum_{\forall x_{1,B}^p} \mu_{1,j_1,B}\left(x_{1,B}^p\right) y_B^p}{\sum_{\forall x_{1,A}^p} \mu_{1,j_1,A}\left(x_{1,A}^p\right) + \sum_{\forall x_{1,B}^p} \mu_{1,j_1,B}\left(x_{1,B}^p\right)} = \frac{\sum_{\forall x_{1,A}^p} \mu_{1,j_1,A}\left(x_{1,A}^p\right) y_A^p + \sum_{\forall x_{1,B}^p} \left[1 - \mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right] y_B^p}{\sum_{\forall x_{1,A}^p} \mu_{1,j_1,A}\left(x_{1,A}^p\right) + \sum_{\forall x_{1,B}^p} \left[1 - \mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]}$$

$$b_{j_1+1} = \frac{\sum_{\forall x_{1,B}^p} \mu_{1,j_1+1,B}\left(x_{1,B}^p\right) y_B^p + \sum_{\forall x_{1,C}^p} \mu_{1,j_1+1,C}\left(x_{1,C}^p\right) y_C^p}{\sum_{\forall x_{1,B}^p} \mu_{1,j_1+1,B}\left(x_{1,B}^p\right) + \sum_{\forall x_{1,C}^p} \mu_{1,j_1+1,C}\left(x_{1,C}^p\right)}$$

4.2.2 The upper limit of b_{j_1}, denoted $\overline{b_{j_1}}$, can be obtained with $\overline{b_{j_1}} = \frac{\sum_{\forall x_{1,B}^p}\left[1-\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]y_B^p}{\sum_{\forall x_{1,B}^p}\left[1-\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]}$, while no data fall in Region A.

4.2.3 The lower limit of b_{j_1+1}, denoted $\underline{b_{j_1+1}}$, can be obtained with $\underline{b_{j_1+1}} = \frac{\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)y_B^p}{\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)}$, while no data samples fall in Region C.

Proposition 4.2: Consider a *monotone* single attribute input data set. By using Equation 4.4, the resulting fuzzy rules are always *monotone* (see Definition 3.4.3).

Proof: We consider any pair of consecutive fuzzy rules, b_{j_1} and b_{j_1+1} for $j_1 = 1,2,\cdots,k_1-1$ using Equation 4.4. If the data set is *monotone*,

$$\underline{b_{j_1+1}} - \overline{b_{j_1}} = \frac{\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)y_B^p}{\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)} - \frac{\sum_{\forall x_{1,B}^p}\left[1-\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]y_B^p}{\sum_{\forall x_{1,B}^p}\left[1-\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]}$$

$$= \frac{\sum_{\forall x_{1,B}^p}\left[1-\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)y_B^p - \sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\sum_{\forall x_{1,B}^p}\left[1-\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]y_B^p}{\left[\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]\left[\sum_{\forall x_{1,B}^p}\left[1-\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]\right]}$$

$$= \frac{\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)y_B^p\sum_{\forall x_{1,B}^p}1 - \sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)y_B^p\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right) - \sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\sum_{\forall x_{1,B}^p}y_B^p + \sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)y_B^p}{\left[\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]\left[\sum_{\forall x_{1,B}^p}\left[1-\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]\right]}$$

$$= \frac{\left[\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)y_B^p\right]\left[\sum_{\forall x_{1,B}^p}1\right] - \left[\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]\left[\sum_{\forall x_{1,B}^p}y_B^p\right]}{\left[\sum_{\forall x_{1,B}^p}\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]\left[\sum_{\forall x_{1,B}^p}\left[1-\mu_{1,j_1+1,B}\left(x_{1,B}^p\right)\right]\right]}$$

Referring to Chebychev's inequality (see (Hardy, Littlewood, & Polya, 1934), p. 43, inequality (2.17.1) for $r = 1$), from which it follows that the numerator is nonnegative.

The denominator is also nonnegative because all μ's are normal and $[1-\mu] \geq 0$. Therefore, $\underline{b_{j_1+1}} \geq \overline{b_{j_1}}$ is always true.

4.3.2 Simulation Studies with Simulated Example 4.1

The generated fuzzy If-Then rules using WM (i.e., Equation 4.3) and Ishibuchi (i.e., Equation 4.4) methods based on the Example 4.1 for $k_1 = 3$ using trapezoidal and triangular fuzzy sets are illustrated in Table 4.1 and Table 4.2, respectively. *Monotone* fuzzy If-Then rules are produced using both methods with trapezoidal and triangular fuzzy sets.

Table 4.1: Generated fuzzy If-Then rules for $k_1 = 3$ using trapezoidal fuzzy set

Methods	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$
WM	0.0682	0.2302	0.3481
Ishibuchi	0.0752	0.2281	0.3431

Table 4.2: Generated fuzzy If-Then rules for $k_1 = 3$ using triangular fuzzy set

Methods	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$
WM	0.0545	0.2279	0.3547
Ishibuchi	0.0797	0.2238	0.3394

The generated fuzzy If-Then rules using WM (i.e., Equation 4.3) and Ishibuchi (i.e., Equation 4.4) methods based on the Example 4.1 for $k_1 = 5$ using trapezoidal and triangular fuzzy sets are illustrated in Table 4.3 and Table 4.4, respectively. Again, *monotone* fuzzy If-Then rules are produced using both methods with trapezoidal and triangular fuzzy sets.

Table 4.3: Generated fuzzy If-Then rules for $k_1 = 5$ using trapezoidal fuzzy set

Methods	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	$A_{1,5}$
WM	0.0371	0.1342	0.2304	0.3090	0.3639
Ishibuchi	0.0417	0.1348	0.2301	0.3082	0.3615

Table 4.4: Generated fuzzy If-Then rules for $k_1 = 5$ using triangular fuzzy set

Methods	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	$A_{1,5}$
WM	0.0263	0.1220	0.2303	0.3168	0.3691
Ishibuchi	0.0397	0.1213	0.2292	0.3156	0.3623

The generated fuzzy If-Then rules using WM (i.e., Equation 4.3) and Ishibuchi (i.e., Equation 4.4) methods based on the Example 4.1 for $k_1 = 7$ using trapezoidal and triangular fuzzy set are illustrated in Table 4.5 and Table 4.6, respectively. Again, *monotone* fuzzy If-Then rules are produced using both methods with trapezoidal and triangular fuzzy sets.

Table 4.5: Generated fuzzy If-Then rules for $k_1 = 7$ using trapezoidal fuzzy set

Methods	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	$A_{1,5}$	$A_{1,6}$	$A_{1,7}$
WM	0.0255	0.0942	0.1660	0.2308	0.2870	0.3342	0.3695
Ishibuchi	0.0286	0.0948	0.1661	0.2306	0.2866	0.3337	0.3680

Table 4.6: Generated fuzzy If-Then rules for $k_1 = 7$ using triangular fuzzy set

Methods	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	$A_{1,5}$	$A_{1,6}$	$A_{1,7}$
WM	0.0174	0.0835	0.1610	0.2307	0.2917	0.3411	0.3732
Ishibuchi	0.0261	0.0822	0.1601	0.2303	0.2906	0.3403	0.3691

4.3.3 Analysis using Multi-attribute *Monotone* Data

Remark 4.3: An analysis of the WM method (Wang, 2003; Wang & Mendel, 1992) for a multi-attribute *monotone* input data (see Definition 4.1.2), $n > 1$, is summarized, as follows.

4.3.1 The lower limit of $b_{\boldsymbol{j}}$, denoted $\underline{b_{\boldsymbol{j}}}$, can be obtained with $\underline{b_{\boldsymbol{j}}} = min_{\forall p:R_{\boldsymbol{j}} \in R^p}(y^p)$, for $\boldsymbol{j} = (j_1, \cdots, j_n)$.

4.3.2 The upper limit of $b_{\boldsymbol{j}}$, denoted $\overline{b_{\boldsymbol{j}}}$, can be obtained with $\overline{b_{\boldsymbol{j}}} = max_{\forall p:R_{\boldsymbol{j}} \in R^p}(y^p)$, for $\boldsymbol{j} = (j_1, \cdots, j_n)$.

Proposition 4.3: Consider a multi-attribute *monotone* input data set. By using Equation 4.3, the resulting fuzzy rules are not always *monotone* (see Definition 3.4.3).

Proof. We consider any pair of consecutive fuzzy rules, $b_{\boldsymbol{j}}$ and $b_{\boldsymbol{j}+\boldsymbol{e}_i}$ for $\boldsymbol{j} = (j_1, \cdots, j_n)$ using Equation 4.3. The upper limit of $b_{\boldsymbol{j}}$, is $\overline{b_{\boldsymbol{j}}}$, where $\overline{b_{\boldsymbol{j}}} = max_{\forall p:R_{\boldsymbol{j}} \in R^p}(y^p)$. The lower limit of $b_{\boldsymbol{j}+\boldsymbol{e}_i}$ is $\underline{b_{\boldsymbol{j}+\boldsymbol{e}_i}}$, where $\underline{b_{\boldsymbol{j}+\boldsymbol{e}_i}} = min_{\forall p:R_{\boldsymbol{j}+\boldsymbol{e}_i} \in R^p}(y^p)$. Therefore, $\underline{b_{\boldsymbol{j}+\boldsymbol{e}_i}} \geq \overline{b_{\boldsymbol{j}}}$ is not always true, even if the data set is *monotone*.

Remark 4.4: An analysis of the Ishibuchi method (Hisao Ishibuchi et al., 1994; Nozaki et al., 1997) for a multi-attribute *monotone* input data (see Definition 4.1.2), $n > 1$, is summarized, as follows.

4.4.1 The lower limit of $b_{\boldsymbol{j}}$, denoted $\underline{b_{\boldsymbol{j}}}$, can be obtained with $\underline{b_{\boldsymbol{j}}} = min_{\{\forall p | \mu_{\boldsymbol{j}}(\boldsymbol{x}^p) > 0\}}(y^p)$ for $\boldsymbol{j} = (j_1, \cdots, j_n)$.

4.4.2 The upper limit of b_j, denoted $\overline{b_j}$, can be obtained with $\overline{b_j} = max_{\{\forall p|\mu_j(x^p)>0\}}(y^p)$ for $\boldsymbol{j} = (j_1, \cdots, j_n)$.

Proposition 4.4: Consider a multi-attribute *monotone* input data set. By using Equation 4.4, the resulting fuzzy rules are not always *monotone* (see Definition 3.4.3).

Proof. We consider any pair of consecutive fuzzy rules, b_j and b_{j+e_i} for $\boldsymbol{j} = (j_1, \cdots, j_n)$ using Equation 4.4. The upper limit of b_j is $\overline{b_j}$, where $\overline{b_j} = max_{\{\forall p|\mu_j(x^p)>0\}}(y^p)$. The lower limit of b_{j+e_i} is $\underline{b_{j+e_i}}$ where $\underline{b_{j+e_i}} = min_{\{\forall p|\mu_{j+e_i}(x^p)>0\}}(y^p)$. Therefore, $\underline{b_{j+e_i}} \geq \overline{b_j}$ is not always true.

4.3.4 Simulation Studies with Simulated Example 4.2

The generated fuzzy If-Then rules using WM (i.e., Equation 4.3) and Ishibuchi (i.e., Equation 4.4) methods based on the Example 4.2 for $k_1 = k_2 = 3$ using trapezoidal fuzzy set are illustrated in Table 4.7 and Table 4.8, respectively. While, the generated fuzzy If-Then rules using WM (i.e., Equation 4.3) and Ishibuchi (i.e., Equation 4.4) methods based on the Example 4.2 for $k_1 = k_2 = 3$ using triangular fuzzy set are illustrated in Table 4.9 and Table 4.10, respectively. *Monotone* and *complete* fuzzy If-Then rules are produced using both methods with trapezoidal and triangular fuzzy sets, respectively.

Table 4.7: Generated fuzzy If-Then rules from WM method for $k_1 = k_2 = 3$ using trapezoidal fuzzy set

$A_{1,3}$	0.4981	0.7376	0.9421
$A_{1,2}$	0.2741	0.6005	0.6813
$A_{1,1}$	0.0541	0.2915	0.4921
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$

Table 4.8: Generated fuzzy If-Then rules from Ishibuchi method for $k_1 = k_2 = 3$ using trapezoidal fuzzy set

$A_{1,3}$	0.4981	0.7376	0.9421
$A_{1,2}$	0.2741	0.6005	0.6813
$A_{1,1}$	0.0541	0.2915	0.4921
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$

Table 4.9: Generated fuzzy If-Then rules from WM method for $k_1 = k_2 = 3$ using triangular fuzzy set

$A_{1,3}$	0.5000	0.7459	0.9583
$A_{1,2}$	0.2722	0.6190	0.6500
$A_{1,1}$	0.0435	0.2845	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$

Table 4.10: Generated fuzzy If-Then rules from Ishibuchi method for $k_1 = k_2 = 3$ using triangular fuzzy set

$A_{1,3}$	0.4954	0.7272	0.9364
$A_{1,2}$	0.2802	0.5790	0.7066
$A_{1,1}$	0.0678	0.2892	0.4894
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$

The generated fuzzy If-Then rules using WM (i.e., Equation 4.3) and Ishibuchi (i.e., Equation 4.4) methods based on the Example 4.2 for $k_1 = k_2 = 5$ using trapezoidal

fuzzy set are illustrated in Table 4.11 and Table 4.12, respectively. While, the generated fuzzy If-Then rules using WM (i.e., Equation 4.3) and Ishibuchi (i.e., Equation 4.4) methods based on the Example 4.2 for $k_1 = k_2 = 5$ using triangular fuzzy set are illustrated in Table 4.13 and Table 4.14, respectively. *Non-monotone* and *incomplete* fuzzy If-Then rules are produced using WM method for trapezoidal and triangular fuzzy set, respectively. While, *non-monotone* and *complete* fuzzy If-Then rules are produced using Ishibuchi method for trapezoidal and triangular fuzzy set, respectively.

Table 4.11: Generated fuzzy If-Then rules from WM method for $k_1 = k_2 = 5$ using trapezoidal fuzzy set

$A_{1,5}$	0.5000	0.5472	0.7327	0.9228	0.9700
$A_{1,4}$	0.4528	0.4778	0.6954	0.8667	0.9450
$A_{1,3}$	0.2623	NaN	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0772	NaN	NaN	0.4869	0.5250
$A_{1,1}$	0.0333	0.0685	0.2623	0.4528	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

Table 4.12: Generated fuzzy If-Then rules from Ishibuchi method for $k_1 = k_2 = 5$ using trapezoidal fuzzy set

$A_{1,5}$	0.5000	0.5543	0.7341	0.9097	0.9655
$A_{1,4}$	0.4382	0.4978	0.6849	0.8304	0.9427
$A_{1,3}$	0.2623	0.3100	0.5409	**0.7400**	**0.6500**
$A_{1,2}$	0.0932	0.3100	0.3100	0.4869	0.5463
$A_{1,1}$	0.0379	0.0912	0.2623	0.4382	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

Table 4.13: Generated fuzzy If-Then rules from WM method for $k_1 = k_2 = 5$ using triangular fuzzy set

$A_{1,5}$	0.5000	0.5463	0.7327	0.9263	0.9725
$A_{1,4}$	0.4538	0.4800	0.6995	0.8733	0.9463
$A_{1,3}$	0.2610	NaN	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0738	NaN	NaN	0.4867	0.5263
$A_{1,1}$	0.0323	0.0677	0.2610	0.4538	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

Table 4.14: Generated fuzzy If-Then rules from Ishibuchi method for $k_1 = k_2 = 5$ using triangular fuzzy set

$A_{1,5}$	0.5000	0.5449	0.7376	0.9220	0.9667
$A_{1,4}$	0.4448	0.4990	0.7075	0.8578	0.9500
$A_{1,3}$	0.2633	0.3105	0.5878	**0.7179**	**0.6500**
$A_{1,2}$	0.0816	0.1130	0.3279	0.4790	0.5375
$A_{1,1}$	0.0385	0.0791	0.2681	0.4479	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

The generated fuzzy If-Then rules using WM (i.e., Equation 4.3) and Ishibuchi (i.e., Equation 4.4) methods based on the Example 4.2 for $k_1 = k_2 = 7$ using trapezoidal fuzzy set are illustrated in Table 4.15 and Table 4.16, respectively. While, the generated fuzzy If-Then rules using WM (i.e., Equation 4.3) and Ishibuchi (i.e., Equation 4.4) methods based on the Example 4.2 for $k_1 = k_2 = 7$ using triangular fuzzy set are illustrated in Table 4.17 and Table 4.18, respectively. *Monotone* and *incomplete* fuzzy If-Then rules are produced using WM method for trapezoidal and triangular fuzzy set, respectively. While, *non-monotone* and *incomplete* fuzzy If-Then rules are produced using Ishibuchi method for trapezoidal fuzzy set and *non-monotone* and *complete* fuzzy If-Then rules are produced using Ishibuchi method for triangular fuzzy set.

Table 4.15: Generated fuzzy If-Then rules from WM method for $k_1 = k_2 = 7$ using trapezoidal fuzzy set

$A_{1,7}$	0.5000	0.5259	0.5857	0.7300	0.8856	0.9459	0.9718
$A_{1,6}$	0.4741	0.5000	NaN	0.7100	0.8371	0.9200	0.9459
$A_{1,5}$	0.3965	0.4600	0.5700	0.6324	0.7641	NaN	NaN
$A_{1,4}$	0.2641	NaN	NaN	NaN	NaN	NaN	NaN
$A_{1,3}$	0.1271	NaN	0.3100	NaN	0.5000	NaN	0.6500
$A_{1,2}$	0.0541	NaN	NaN	NaN	0.4600	0.5000	0.5259
$A_{1,1}$	0.0326	0.0541	0.1346	0.2641	0.3965	0.4741	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,6}$

Table 4.16: Generated fuzzy If-Then rules from Ishibuchi method for $k_1 = k_2 = 7$ using trapezoidal fuzzy set

$A_{1,7}$	0.5000	0.5268	**0.5857**	0.7314	0.8856	0.9473	0.9718
$A_{1,6}$	0.4732	0.4977	**0.5939**	0.7221	0.8406	0.9250	0.9505
$A_{1,5}$	0.3965	0.4391	0.5700	0.6418	0.7641	0.7971	NaN
$A_{1,4}$	0.2627	0.2686	0.3100	0.5643	0.7271	**0.7600**	**0.6500**
$A_{1,3}$	0.1271	0.1329	0.3100	0.3100	0.5000	0.5000	0.6500
$A_{1,2}$	0.0527	0.0592	0.1700	0.2686	0.4428	0.4968	0.5213
$A_{1,1}$	0.0326	0.0513	0.1346	0.2627	0.3965	0.4755	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,6}$

Table 4.17: Generated fuzzy If-Then rules from WM method for $k_1 = k_2 = 7$ using triangular fuzzy set

$A_{1,7}$	0.5000	0.5214	0.5700	NaN	0.9100	0.9586	0.9800
$A_{1,6}$	0.4786	0.5000	0.5942	0.7186	0.8530	0.9372	0.9586
$A_{1,5}$	0.3958	0.4292	0.5700	0.6358	0.7702	NaN	NaN
$A_{1,4}$	0.2600	0.2700	NaN	NaN	NaN	NaN	NaN
$A_{1,3}$	0.1199	0.1299	0.3100	NaN	0.5000	NaN	0.6500
$A_{1,2}$	0.0414	0.0545	0.1700	0.2700	0.4292	0.5000	0.5214
$A_{1,1}$	NaN	0.0414	0.1199	0.2600	0.3958	0.4786	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,6}$

Table 4.18: Generated fuzzy If-Then rules from Ishibuchi method for $k_1 = k_2 = 7$ using triangular fuzzy set

$A_{1,7}$	0.5000	0.5275	0.5708	0.7322	0.9029	0.9498	0.9743
$A_{1,6}$	0.4725	0.4968	0.5688	0.7248	0.8557	0.9288	0.9558
$A_{1,5}$	0.4081	0.4392	0.5395	0.6593	0.7816	0.8656	0.9472
$A_{1,4}$	0.2606	0.2678	0.3100	0.5499	0.7301	**0.7600**	**0.6500**
$A_{1,3}$	0.1139	0.1210	0.3100	0.3100	0.4923	0.5017	0.6104
$A_{1,2}$	0.0502	0.0583	0.1421	0.2678	0.4442	0.4941	0.5185
$A_{1,1}$	0.0317	0.0484	0.1162	0.2606	0.4081	0.4754	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,6}$

4.4 A Framework for Developing *Monotone* TSK FIS models with *Ad Hoc* methods

Our analytical and empirical studies demonstrate that fuzzy rules generated by the WM and Ishibuchi methods are not always *monotone*, even if the data set are *monotone*. Besides that, fuzzy rules can be *incomplete* too. This implies that fuzzy rules generated by the WM and Ishibuchi methods need to be pre-processed for FIS modelling in practice. In this section, a framework for practical modelling of *monotone* TSK FIS models with both *ad hoc* methods is proposed. The framework consists of six stages, as presented in Figure 4.17.

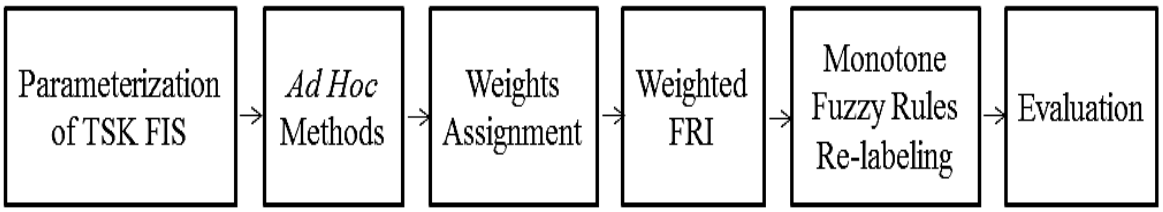

Figure 4.17: The proposed framework for practical modelling of *monotone* TSK FIS with *ad hoc* methods

Each stage of the proposed framework is detailed, as follows.

Stage 1: *Parameterization of TSK FIS*

Firstly, specify the parameters of the TSK FIS model, i.e., $f(\boldsymbol{x}, A_j; \boldsymbol{b_j})$. A strong partition is used. Note that A_j in the input space is pre-determined, and A_j can be determined manually or using a meta-heuristic procedure. In addition, $\boldsymbol{b_j} = (b_{1,\cdots,1}, b_{1,\cdots,2}, \cdots, b_{k_1,\cdots,k_n})$ is a vector that consists of all b_j where $\boldsymbol{j} = (j_1, \cdots, j_n)$.

Stage 2: *Ad hoc Methods*

Equation 4.3 or 4.4 with respect to the WM or Ishibuchi method is used to learn the fuzzy rules and determine $\boldsymbol{b_j}$. Note that $\boldsymbol{b_j}$ can be *incomplete* or *non-monotone.* In regard to Example 4.2, $\boldsymbol{b_j}$ as in Table 4.11 and Table 4.13, is obtained using WM method using trapezoidal and triangular fuzzy set, respectively. *Incomplete* and *non-monotone* fuzzy rules are obtained.

Stage 3: *Weight Assignment*

Four strategies to assign weights, (w_j), to R_j: $A_j \rightarrow b_j$ are considered, as follows.

A. A one-zero strategy for *known-unknown* fuzzy rules.

$$w_j^{\text{One-Zero}} = \begin{cases} 1, & if \;\; b_j \;\; is\; known \\ 0, & if \;\; b_j \; is\; unknown \end{cases} \tag{4.5.a}$$

B. The number of data samples over R_j, i.e., w_j^{data}, can be obtained using (4.5.b) or (4.5.c), for the WM or Ishibuchi method, respectively.

$$w_j^{\text{data,WM}} = \sum_{\forall p: R_j \in R^p} (1) \tag{4.5.b}$$

$$w_j^{\text{data,Ishibuchi}} = \sum_{\forall p | \mu_j(\boldsymbol{x}^p) > 0} (1) \tag{4.5.c}$$

C. Summation of the *activation degree*, i.e., w_j^{sum}, can be obtained using (4.5.d) or (4.5.e), for the WM or Ishibuchi method, respectively.

$$w_j^{\text{sum,WM}} = \sum_{\forall p: R_j \in R^p} \mu_j(\boldsymbol{x}^p) \tag{4.5.d}$$

$$w_j^{\text{sum,Ishibuchi}} = \sum_{\forall p} \mu_j(\boldsymbol{x}^p) \tag{4.5.e}$$

D. The highest *activation degree*, i.e., $w_j^{\max}$ can be obtained using (4.5.f) or (4.5.g) for the WM or Ishibuchi method, respectively.

$$w_j^{\text{max,WM}} = max_{\forall p: R_j \in R^p} \left(\mu_j(\boldsymbol{x}^p)\right) \tag{4.5.f}$$

$$w_j^{\text{max,Ishibuchi}} = max_{\forall p} \left(\mu_j(\boldsymbol{x}^p)\right) \tag{4.5.g}$$

Note that if b_j is *unknown*, $w_j^{\text{One-Zero}}$, w_j^{data}, w_j^{sum}, $w_j^{\max}$ all are equal to 0. The weights obtained using Equation 4.5 can be normalized, i.e., $w_j^{\text{normalized}} = \frac{w_j - min\{\boldsymbol{w_j}\}}{max\{\boldsymbol{w_j}\} - min\{\boldsymbol{w_j}\}} \in [0,1]$, where $\boldsymbol{w_j} = \left(w_{1,\cdots,1}, w_{1,\cdots,2}, \cdots, w_{k_1,\cdots,k_n}\right)$.

Stage 4: *Weighted Fuzzy Rule Interpolation (WFRI)*

A modified version of the linear Weighted Fuzzy Rule Interpolation (Baranyi et al., 2004; Zhiheng Huang & Shen, 2006; Kóczy & Hirota, 1993) (WFRI) procedure for zero-

order fuzzy If-Then rules is used. The aim is to deduce the missing fuzzy If-Then rules, i.e., for the case b_j is *unknown*, using fuzzy rules obtained from the WM or Ishibuchi method. The missing fuzzy If-Then rules are denoted $R_j^{\#}: A_j^{\#} \rightarrow b_j^{\#}$. The fuzzy rules deduced from the WM or Ishibuchi methods are denoted $R_j^{*}: A_j^{*} \rightarrow b_j^{*}$. A combination of all $b_j^{\#}$(i.e., $\boldsymbol{b}_j^{\#}$) and all b_j^{*}(i.e., $\boldsymbol{b}_j^{*}$) form $\boldsymbol{b}_j$, i.e., $\boldsymbol{b}_j = \left(\boldsymbol{b}_j^{\#}, \boldsymbol{b}_j^{*}\right)$.

The degree of closeness (a distance similarity measure) between $A_j^{\#}$ and A_j^{*} is measured using $S\left(A_j^{\#}, A_j^{*}\right) = \frac{1}{d\left(A_j^{\#}, A_j^{*}\right)}$, where d is the Euclidean distance between the cores of $A_j^{\#}$ and A_j^{*}, in which the membership degree is 1. Note that $d\left(A_j^{\#}, A_j^{*}\right)$ is defined by $d\left(A_j^{\#}, A_j^{*}\right) = \sqrt{\sum_{i=1}^{n}(A_{i,j}^{\#,c} - A_{i,j}^{*,c})^2}$, where $j = 1, \cdots, k_i$, $A_{i,j}^{\#,c}$ and $A_{i,j}^{*,c}$ are the core values for $A_{i,j}^{\#}$ and $A_{i,j}^{*}$, respectively, in the X_i domain. Then, $b_j^{\#}$ is obtained using Equation 4.6

$$b_j^{\#} = \frac{\sum_{\forall A_j^{*}}\left[S\left(A_j^{\#}, A_j^{*}\right) \times b_j^{*} \times w_j\right]}{\sum_{\forall A_j^{*}} S\left(A_j^{\#}, A_j^{*}\right)} \tag{4.6}$$

where w_j can be determined using Equation 4.5 in Stage 3, with or without normalization.

Stage 5: *Monotone Fuzzy Rules Relabeling (MFRR)*

Fuzzy rules obtained from Stage 4, i.e., $\boldsymbol{b}_j = \left(\boldsymbol{b}_j^{\#}, \boldsymbol{b}_j^{*}\right)$, are re-labelled using a nonlinear mathematical programming procedure. The re-labelled outcome is denoted $\boldsymbol{b}_j^{\mathrm{rel}} = \left(b_{1,\cdots,1}^{\mathrm{rel}}, b_{1,\cdots,2}^{\mathrm{rel}}, \cdots, b_{k_1,\cdots,k_n}^{\mathrm{rel}}\right) \in \mathbb{R}^M$. The objective function of $\boldsymbol{b}_j^{\mathrm{rel}}$ to be minimized, i.e., diff, is as follows:

$$\text{diff}\left(\boldsymbol{b}_{\boldsymbol{j}}^{\text{rel}}\right) = \sum\nolimits_{\forall A_j} \left(w_{\boldsymbol{j}}\left(\, b_{\boldsymbol{j}} - b_{\boldsymbol{j}}^{\text{rel}}\right)\right)^2 \tag{4.7.a}$$

where $w_{\boldsymbol{j}}$ can be determined using Equation 4.5 in Stage 3, with or without normalization, diff: $\mathbb{R}^M \rightarrow \mathbb{R}$, and subject to L inequality constraints, as follows.

$$g_{lo}\left(\boldsymbol{b}_{\boldsymbol{j}}^{\text{rel}}\right) = -\left(b_{\boldsymbol{j}+\boldsymbol{e}_i}^{\text{rel}} - b_{\boldsymbol{j}}^{\text{rel}}\right) \leq 0 \tag{4.7.b}$$

where $lo = 1,2,\cdots,L$ and $g_{lo}\colon \mathbb{R}^M \rightarrow \mathbb{R}^L$. $diff\left(\boldsymbol{b}_{\boldsymbol{j}}^{\text{rel}}\right)$ and $g_{lo}\left(\boldsymbol{b}_{\boldsymbol{j}}^{\text{rel}}\right)$ are continuous functions with differentiable second partial derivatives. The Lagrangian function corresponding to Equation 4.7 is given by

$$\mathcal{L}\left(\boldsymbol{b}_{\boldsymbol{j}}^{\text{rel}}, \boldsymbol{\lambda}\right) = diff\left(\boldsymbol{b}_{\boldsymbol{j}}^{\text{rel}}\right) + \sum_{lo=1}^{L} \lambda_{lo} g_{lo}\left(\boldsymbol{b}_{\boldsymbol{j}}^{\text{rel}}\right) \tag{4.8}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_L)$ denotes the vector of Lagrange multipliers, with $\boldsymbol{\lambda} \in \mathbb{R}^L$. We develop an algorithm for undertaking the constrained minimization problem i.e., Equation 4.7. The algorithm consists of a sequence of unconstrained minimization problems that need to be solved at each iteration. Equations 4.7.a and 4.7.b are transformed to 4.9, as follows.

$$F\left(\boldsymbol{b}_{\boldsymbol{j}}^{\text{rel}}, \boldsymbol{\lambda}\right) = \text{diff}\left(\boldsymbol{b}_{\boldsymbol{j}}^{\text{rel}}\right) + \sum_{lo=1}^{L} \lambda_{lo}\alpha_{lo} + \sum_{lo=1}^{L} \alpha_{lo}^2 \tag{4.9}$$

where $\alpha_{lo} = \max\{g_{lo}(\boldsymbol{b}_j^{\text{rel}}), -\frac{\lambda_{lo}}{2}\}$, $\lambda_{lo}(lo = 1,2,\cdots,L)$ are the Lagrange multipliers, and $\boldsymbol{b}_j^{\text{rel}} \in \mathbb{R}^M$. The third term, or the quadratic penalty-like term, in Equation 4.9, i.e., $\sum_{lo=1}^{L} \alpha_{lo}^2$, is used to guarantee that the deduced outcome of $\boldsymbol{b}_j^{\text{rel}}$ from Equation 4.7.a not only is a stationary point, but also a minimum of $F(\boldsymbol{b}_j^{\text{rel}}, \boldsymbol{\lambda})$ in (14) with the associated $\boldsymbol{\lambda}$ setting. In Algorithm 1, a quasi-newton updating scheme, i.e., the Davidon-Fletcher-Powell (DFP) method (Davidon, 1991; Fletcher & Powell, 1963), is chosen to solve $\boldsymbol{b}_j^{\text{rel},(r+1)}$ (Step 3 of Algorithm 1), for a fixed $\boldsymbol{\lambda}$ setting at the r-th iteration. Notice that $\boldsymbol{\lambda}$ is updated in Step 4 of Algorithm 1, where $\boldsymbol{\alpha}^{(\mathbf{r})} = \left(\alpha_1^{(\mathbf{r})}, \cdots, \alpha_L^{(\mathbf{r})}\right)$, until the convergence criterion (Step 6 of Algorithm 1) is satisfied.

For the DFP method in Step 3 of Algorithm 1, the partial derivatives of the *F* function are estimated by using the central finite difference method, i.e., a numerical method based on mathematical discretization of differential equations, as defined in Equation 4.10

$$\frac{\partial F}{\partial b_{1,\cdots,k_n}^{\text{rel},(r)}} = \frac{F\left(\boldsymbol{b}_j^{\text{rel},(r),1}, \boldsymbol{\lambda}^{(r)}\right) - F\left(\boldsymbol{b}_j^{\text{rel},(r),2}, \boldsymbol{\lambda}^{(r)}\right)}{2\Delta} \tag{4.10}$$

where $\boldsymbol{b}_j^{\text{rel},(r),1} = \left(b_{1,\cdots,1}^{\text{rel},(r)}, \cdots, b_{1,\cdots,k_n}^{\text{rel},(r)} + \Delta, \cdots, b_{k_1,\cdots,k_n}^{\text{rel},(r)}\right)$; $\boldsymbol{b}_j^{\text{rel},(r),2} = \left(b_{1,\cdots,1}^{\text{rel},(r)}, \cdots, b_{1,\cdots,k_n}^{\text{rel},(r)} - \Delta, \cdots, b_{k_1,\cdots,k_n}^{\text{rel},(r)}\right)$. Note that Δ is the limit in Equation 4.10, together with ε_1 and ε_2 that need to be pre-specified. An initial search interval for the direct search method, i.e., $\left[\underline{\beta}, \overline{\beta}\right]$, is set to $= \left[min\{\boldsymbol{b}_j\}, max\{\boldsymbol{b}_j\}\right]$. A golden ratio (Kaufman, 1998), $\tau = \frac{\sqrt{5}-1}{2} = 0.618$, is used as part of the direct search method in Step 3 of Algorithm 1.

Algorithm 1: The MFRR-based method

Input: $\boldsymbol{b_j}$

Output: $\boldsymbol{b_j^{\mathrm{rel}}} = \boldsymbol{b_j^{\mathrm{rel,(r)}}}$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(\mathrm{r})}$

1. Set r = 1, initialize $\boldsymbol{b_j^{\mathrm{rel,(1)}}} = \boldsymbol{b_j}$, $\boldsymbol{\lambda}^{(1)} = 0$, $\boldsymbol{G}^{(1)} = \boldsymbol{I}$, $\varepsilon_1, \varepsilon_2$, Δ, $\left[\underline{\beta}, \overline{\beta}\right]$
2. **repeat**
3. $\boldsymbol{b_j^{\mathrm{rel,(r+1)}}} \in \arg\min_{\boldsymbol{b_j^{\mathrm{rel,(r)}}}} F\left(\boldsymbol{b_j^{\mathrm{rel,(r)}}}, \boldsymbol{\lambda}^{(\mathrm{r})}\right)$
4. $\boldsymbol{\lambda}^{(\mathrm{r+1})} = \boldsymbol{\lambda}^{(\mathrm{r})} + 2\boldsymbol{\alpha}^{(\mathrm{r})}$
5. r ← r + 1
6. **until** $\left|F\left(\boldsymbol{b_j^{\mathrm{rel,(r)}}}, \boldsymbol{\lambda}^{(\mathrm{r})}\right) - F(\boldsymbol{b_j^{\mathrm{rel,(r-1)}}}, \boldsymbol{\lambda}^{(\mathrm{r-1})})\right| < \varepsilon_1$

It is worth noting that other constrained mathematical programming methods, e.g., sequential quadratic programming (Boggs & Tolle, 1995) and meta-heuristic methods, e.g., Genetic Algorithm (Goldberg, 1989) and Harmony Search (Geem, Kim, & Loganathan, 2001), can also be used to solve the problem, i.e., Equation 4.7.

Stage 6: *Evaluation*

Accuracy of the generated FIS model using Equation 4.3 or Equation 4.4 is validated by using the Sum Square Error (SSE), i.e., Equation 4.11. In this evaluation, the data set used is denoted $(\boldsymbol{x}^v; y^v), v = 1,2,\cdots,V$. Note that $(\boldsymbol{x}^v; y^v)$ can be a training data set or a validation data set. In practice, the FIS model with the lowest SSE value from a weight assignment strategy is chosen as the final candidate.

$$\frac{\partial F}{\partial b_{1,\cdots,k_n}^{\mathrm{rel,(r)}}} = \frac{F\left(\boldsymbol{b_j^{\mathrm{rel,(r),1}}}, \boldsymbol{\lambda}^{(\mathrm{r})}\right) - F\left(\boldsymbol{b_j^{\mathrm{rel,(r),2}}}, \boldsymbol{\lambda}^{(\mathrm{r})}\right)}{2\Delta} \tag{4.11}$$

4.4.1 Simulation Studies with Simulated Example 4.2

Example 4.2 is re-visited. For $k_1 = k_2 = 5$, the weights obtained for the WM and Ishibuchi methods using trapezoidal fuzzy sets after Stage 3 are summarized in Table 4.19 (a) and (b), respectively. Columns "j_1" and "j_2" represent the labels of membership functions of X_1 and X_2, respectively. Columns "$w_{j_1,j_2}^{\text{One-Zero}}$", "$w_{j_1,j_2}^{\text{data}}$", "$w_{j_1,j_2}^{\text{sum}}$", "$w_{j_1,j_2}^{\max}$", summarize the weights obtained using Equation 4.5. For the WM method, $j_1 = 2$ and $j_2 = 1$, i.e., $b_{2,1} = 0.0772$, the obtained weights are: $w_{2,1}^{\text{One-Zero}} = 1$, $w_{2,1}^{\text{data}} = 4$, $w_{2,1}^{\text{sum}} = 3.6$ and $w_{2,1}^{\max} = 1$, respectively. For $j_1 = 2$ and $j_2 = 2$, $b_{2,2}$ is *unknown*, resulting in $w_{2,2}^{\text{One-Zero}} = 0$, $w_{2,2}^{\text{data}} = 0$, $w_{2,2}^{\text{sum}} = 0$ and $w_{2,2}^{\max} = 0$, respectively.

Table 4.19: The values of $\boldsymbol{b_j}$ and weights for $k_1 = k_2 = 5$ using trapezoidal fuzzy set after Stage 3 (a) using WM method and (b) using Ishibuchi method

Using WM method						
j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$
1	1	0.0333	1	3	3.0000	1.00
1	2	0.0685	1	3	2.6000	1.00
1	3	0.2623	1	6	4.4001	1.00
1	4	0.4528	1	4	3.6000	1.00
1	5	0.5000	1	4	4.0000	1.00
2	1	0.0772	1	4	3.6000	1.00
2	2	NaN	0	0	0.0000	0.00
2	3	NaN	0	0	0.0000	0.00
2	4	0.4869	1	3	2.4400	1.00
2	5	0.5250	1	2	1.6000	0.80
3	1	0.2623	1	6	4.4001	1.00
3	2	NaN	0	0	0.0000	0.00
3	3	0.5409	1	3	1.3201	0.60
3	4	0.7600	1	1	0.6000	0.60
3	5	0.6500	1	1	0.6000	0.60
4	1	0.4528	1	4	3.6000	1.00
4	2	0.4778	1	2	1.4400	0.80
4	3	0.6954	1	5	3.4801	1.00
4	4	0.8667	1	3	2.4400	1.00
4	5	0.945	1	2	1.6000	0.80
5	1	0.5000	1	3	3.0000	1.00
5	2	0.5472	1	4	3.6000	1.00
5	3	0.7327	1	3	2.2001	1.00
5	4	0.9228	1	4	3.6000	1.00
5	5	0.9700	1	4	4.0000	1.00

(a)

Table 4.19 continued

Using Ishibuchi method						
j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$
1	1	0.0379	1	7	3.7999	1.00
1	2	0.0912	1	5	3.4000	1.00
1	3	0.2623	1	6	4.4001	1.00
1	4	0.4382	1	8	4.7599	1.00
1	5	0.5000	1	9	4.8399	1.00
2	1	0.0932	1	6	4.4000	1.00
2	2	0.3100	1	1	0.1600	0.16
2	3	0.3100	1	1	0.2400	0.24
2	4	0.4869	1	3	2.4400	1.00
2	5	0.5463	1	4	2.1600	0.80
3	1	0.2623	1	6	4.4001	1.00
3	2	0.3100	1	1	0.2400	0.24
3	3	0.5409	1	3	1.3201	0.60
3	4	0.7400	1	2	0.8400	0.60
3	5	0.6500	1	1	0.6000	0.60
4	1	0.4382	1	8	4.7599	1.00
4	2	0.4978	1	3	1.8400	0.80
4	3	0.6849	1	7	4.1201	1.00
4	4	0.8304	1	7	3.7199	1.00
4	5	0.9427	1	3	1.7600	0.80
5	1	0.5000	1	8	3.8399	1.00
5	2	0.5543	1	6	4.1600	1.00
5	3	0.7341	1	5	2.5201	1.00
5	4	0.9097	1	8	4.4400	1.00
5	5	0.9655	1	9	4.8399	1.00

(b)

There are 22 fuzzy rules generated from WM method (i.e., Equation 4.3) using trapezoidal fuzzy set (see Table 4.11) with 3 missing fuzzy rules, i.e., $R_{2,2}^{\#}$, $R_{2,3}^{\#}$ and $R_{3,2}^{\#}$ and there are 25 fuzzy rule generated from Ishibuchi method (i.e., Equation 4.4) using trapezoidal fuzzy set (see Table 4.12). The missing fuzzy rules for WM method are obtained using Equation 4.6 with different weight assignment strategies, along with weight

normalization. The obtained outcome for WM method after Stage 4 is summarized in Table 4.20. As depicted in Table 4.20 (b), with $w_{j_1,j_2}^{\text{data}}$, for $A_{j_1=2,j_2=2}^{\#}$, $b_{2,2}^{\#} = 0.2361$, $A_{j_1=2,j_2=3}^{\#}$, $b_{2,3}^{\#} = 0.2595$, and $A_{j_1=3,j_2=2}^{\#}$, $b_{3,2}^{\#} = 0.2692$ are obtained, respectively.

Table 4.20: Fuzzy rules for $k_1 = k_2 = 5$ after Stage 4 for trapezoidal fuzzy set using WM method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$					
$A_{1,5}$	0.5000	0.5472	0.7327	0.9228	0.9700
$A_{1,4}$	0.4528	**0.4778**	0.6954	0.8667	0.9450
$A_{1,3}$	0.2623	**0.4942**	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0772	0.4325	**0.4946**	**0.4869**	0.5250
$A_{1,1}$	0.0333	0.0685	0.2623	0.4528	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(a)

With $w_{j_1,j_2}^{\text{data}}$					
$A_{1,5}$	0.5000	0.5472	0.7327	0.9228	0.9700
$A_{1,4}$	0.4528	0.4778	0.6954	0.8667	0.9450
$A_{1,3}$	0.2623	0.2692	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0772	0.2361	**0.2595**	0.4869	0.5250
$A_{1,1}$	0.0333	0.0685	**0.2623**	0.4528	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(b)

With w_{j_1,j_2}^{sum}					
$A_{1,5}$	0.5000	0.5472	0.7327	0.9228	0.9700
$A_{1,4}$	0.4528	0.4778	0.6954	0.8667	0.9450
$A_{1,3}$	0.2623	0.2878	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0772	0.2556	0.2783	0.4869	0.5250
$A_{1,1}$	0.0333	0.0685	0.2623	0.4528	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(c)

Table 4.20 continued

With $w_{j_1,j_2}^{\max}$					
$A_{1,5}$	0.5000	0.5472	0.7327	0.9228	0.9700
$A_{1,4}$	0.4528	0.4778	0.6954	0.8667	0.9450
$A_{1,3}$	0.2623	0.4384	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0772	0.3852	0.4331	0.4869	0.5250
$A_{1,1}$	0.0333	0.0685	0.2623	0.4528	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(d)

Then, for the *complete* and *non-monotone* fuzzy If-Then rules with WM method obtained after Stage 4 using trapezoidal fuzzy set (see Table 4.20) and the *complete* and *non-monotone* fuzzy If-Then rules with Ishibuchi method obtained after Stage 2 using trapezoidal fuzzy set (see Table 4.12) are re-labelled at Stage 5. The re-labelled fuzzy If-Then rules, i.e., $\boldsymbol{b}_j^{\text{rel}}$ **,** using Algorithm 1 with four weight assignment strategies along and weight normalization for WM and Ishibuchi methods using trapezoidal fuzzy set are presented in Table 4.21 and Table 4.22, respectively.

For WM method, two termination thresholds, i.e., $\varepsilon_1 = 10^{-12}$,$\varepsilon_2 = 10^{-12}$, an initial search interval for the direct search method, i.e., $\left[\underline{\beta}, \overline{\beta}\right] = [0.0333, 0.9700]$, and the limit of the central finite difference method, i.e., $\Delta = 0.001$, are used. For Ishibuchi method, two termination thresholds, i.e., $\varepsilon_1 = 10^{-11}$,$\varepsilon_2 = 10^{-10}$, an initial search interval for the direct search method, i.e., $\left[\underline{\beta}, \overline{\beta}\right] = [0.0379, 0.9655]$, and the limit of the central finite difference method, i.e., $\Delta = 0.001$, are used.

Table 4.21: Fuzzy rules for $k_1 = k_2 = 5$ after Stage 5 for trapezoidal fuzzy set using WM method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

	With $w_{j_1,j_2}^{\text{One-Zero}}$				
$A_{1,5}$	0.5000	0.5472	0.7327	0.9228	0.9700
$A_{1,4}$	0.4528	0.4778	0.6954	0.8667	0.9450
$A_{1,3}$	0.2623	0.4755	0.5409	0.7050	0.7050
$A_{1,2}$	0.0772	0.4325	0.4858	0.4869	0.5250
$A_{1,1}$	0.0333	0.0685	0.2623	0.4528	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(a)

	With $w_{j_1,j_2}^{\text{data}}$				
$A_{1,5}$	0.5000	0.5472	0.7327	0.9228	0.9700
$A_{1,4}$	0.4528	0.4778	0.6954	0.8667	0.9450
$A_{1,3}$	0.2623	0.2692	0.5409	0.7050	0.7050
$A_{1,2}$	0.0772	0.2361	0.2635	0.4869	0.5250
$A_{1,1}$	0.0333	0.0685	0.2623	0.4528	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(b)

	With w_{j_1,j_2}^{sum}				
$A_{1,5}$	0.5000	0.5472	0.7327	0.9228	0.9700
$A_{1,4}$	0.4528	0.4778	0.6954	0.8667	0.9450
$A_{1,3}$	0.2623	0.2878	0.5409	0.7050	0.7050
$A_{1,2}$	0.0772	0.2556	0.2783	0.4869	0.5250
$A_{1,1}$	0.0333	0.0685	0.2623	0.4528	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(c)

Table 4.21 continued

With $w_{j_1,j_2}^{\max}$					
$A_{1,5}$	0.5000	0.5472	0.7327	0.9228	0.9700
$A_{1,4}$	0.4528	0.4778	0.6954	0.8667	0.9450
$A_{1,3}$	0.2623	0.4384	0.5409	0.7050	0.7050
$A_{1,2}$	0.0772	0.3852	0.4331	0.4869	0.5250
$A_{1,1}$	0.0333	0.0685	0.2623	0.4528	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(d)

Table 4.22: Fuzzy rules for $k_1 = k_2 = 5$ after Stage 5 for trapezoidal fuzzy set using Ishibuchi method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$					
$A_{1,5}$	0.5000	0.5543	0.7341	0.9097	0.9655
$A_{1,4}$	0.4382	0.4978	0.6849	0.8304	0.9427
$A_{1,3}$	0.2623	0.3101	0.5409	0.6950	0.6950
$A_{1,2}$	0.0932	0.3099	0.3101	0.4869	0.5463
$A_{1,1}$	0.0379	0.0912	0.2623	0.4382	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(a)

With $w_{j_1,j_2}^{\text{data}}$					
$A_{1,5}$	0.5000	0.5543	0.7341	0.9097	0.9655
$A_{1,4}$	0.4382	0.4978	0.6849	0.8304	0.9427
$A_{1,3}$	0.2623	0.3214	0.5409	0.7400	0.7572
$A_{1,2}$	0.0932	0.2869	0.3214	0.4869	0.5463
$A_{1,1}$	0.0379	0.0912	0.2623	0.4382	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(b)

Table 4.22 continued

With w_{j_1,j_2}^{sum}					
$A_{1,5}$	0.5000	0.5543	0.7341	0.9097	0.9655
$A_{1,4}$	0.4382	0.4978	0.6849	0.8304	0.9427
$A_{1,3}$	0.2623	0.3127	0.5409	0.7160	0.7161
$A_{1,2}$	0.0932	0.3040	0.3127	0.4869	0.5463
$A_{1,1}$	0.0379	0.0912	0.2623	0.4382	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(c)

With $w_{j_1,j_2}^{\max}$					
$A_{1,5}$	0.5000	0.5543	0.7341	0.9097	0.9655
$A_{1,4}$	0.4382	0.4978	0.6849	0.8304	0.9427
$A_{1,3}$	0.2623	0.3103	0.5409	0.6950	0.6950
$A_{1,2}$	0.0932	0.3094	0.3103	0.4869	0.5463
$A_{1,1}$	0.0379	0.0912	0.2623	0.4382	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(d)

After Stage 5, *complete and monotone* fuzzy If-Then rules are produced for both methods with trapezoidal fuzzy set. The obtained difference scores using WM method for $w_{j_1,j_2}^{\text{One-Zero}}, w_{j_1,j_2}^{\text{data}}$, w_{j_1,j_2}^{sum}, $w_{j_1,j_2}^{\max}$ are 0.0061, 1.6857e-04, 1.1339e-04, and 0.0022, respectively. While, the obtained difference scores using Ishibuchi method for $w_{j_1,j_2}^{\text{One-Zero}}, w_{j_1,j_2}^{\text{data}}$, w_{j_1,j_2}^{sum}, $w_{j_1,j_2}^{\max}$ are 0.0041, 1.0000e-06, 5.1771e-05, and 0.0011, respectively.

For $k_1 = k_2 = 5$, the weights obtained for the WM and Ishibuchi methods using triangular fuzzy sets after Stage 3 are summarized in Table 4.23 (a) and (b), respectively. For WM method, $j_1 = 2$ and $j_2 = 1$, i.e., $b_{2,1} = 0.0738$, the obtained weights are:

$w_{2,1}^{\text{One-Zero}} = 1$, $w_{2,1}^{\text{data}} = 4$, $w_{2,1}^{\text{sum}} = 2.56$ and $w_{2,1}^{\text{max}} = 0.8$, respectively. For $j_1 = 2$ and $j_2 = 2$, $b_{2,2}$ is *unknown*, resulting in $w_{2,2}^{\text{One-Zero}} = 0$, $w_{2,2}^{\text{data}} = 0$, $w_{2,2}^{\text{sum}} = 0$ and $w_{2,2}^{\text{max}} = 0$, respectively.

Table 4.23: The values of $\boldsymbol{b_j}$ and weights for $k_1 = k_2 = 5$ using triangular fuzzy set after Stage 3 (a) using WM method and (b) using Ishibuchi method

Using WM method						
j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	w_{j_1,j_2}^{max}
1	1	0.0323	1	3	1.56	0.60
1	2	0.0677	1	3	2.08	0.80
1	3	0.2610	1	6	3.52	1.00
1	4	0.4538	1	4	2.56	0.80
1	5	0.5000	1	4	2.56	1.00
2	1	0.0738	1	4	2.56	0.80
2	2	NaN	0	0	0	0
2	3	NaN	0	0	0	0
2	4	0.4867	1	3	1.92	0.64
2	5	0.5263	1	2	1.28	0.80
3	1	0.2610	1	6	3.52	1.00
3	2	NaN	0	0	0	0
3	3	0.5409	1	3	1.32	0.60
3	4	0.7600	1	1	0.48	0.48
3	5	0.6500	1	1	0.60	0.60
4	1	0.4538	1	4	2.56	0.80
4	2	0.4800	1	2	1.28	0.64
4	3	0.6995	1	5	3.04	0.80
4	4	0.8733	1	3	1.92	0.64
4	5	0.9463	1	2	1.28	0.80
5	1	0.5000	1	3	2.20	1.00
5	2	0.5463	1	4	2.56	0.80
5	3	0.7327	1	3	1.32	0.60
5	4	0.9263	1	4	2.56	0.80
5	5	0.9725	1	4	2.56	1.00

(a)

Table 4.23 continued

Using Ishibuchi method						
j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$
1	1	0.0385	1	7	2.20	0.60
1	2	0.0791	1	8	3.40	0.80
1	3	0.2681	1	10	4.08	1.00
1	4	0.4479	1	11	4.24	0.80
1	5	0.5000	1	9	3.24	1.00
2	1	0.0816	1	9	3.92	0.80
2	2	0.1130	1	7	1.60	0.32
2	3	0.3279	1	7	1.52	0.40
2	4	0.4790	1	8	3.20	0.64
2	5	0.5375	1	7	2.56	0.80
3	1	0.2633	1	11	4.20	1.00
3	2	0.3105	1	8	1.52	0.40
3	3	0.5878	1	9	1.96	0.60
3	4	0.7179	1	5	1.12	0.48
3	5	0.6500	1	1	0.60	0.60
4	1	0.4448	1	10	4.00	0.80
4	2	0.4990	1	9	3.20	0.64
4	3	0.7075	1	15	5.12	0.80
4	4	0.8578	1	12	4.32	0.64
4	5	0.9500	1	6	2.16	0.80
5	1	0.5000	1	8	2.88	1.00
5	2	0.5449	1	8	3.44	0.80
5	3	0.7376	1	10	2.32	0.60
5	4	0.9220	1	11	3.92	0.80
5	5	0.9667	1	9	3.24	1.00

(b)

There are 22 fuzzy rules generated from WM method (i.e., Equation 4.3) using triangular fuzzy set (see Table 4.13) with 3 missing fuzzy rules, i.e., $R_{2,2}^{\#}$, $R_{2,3}^{\#}$ and $R_{3,2}^{\#}$ and there are 25 fuzzy rule generated from Ishibuchi method (i.e., Equation 4.4) using triangular fuzzy set (see Table 4.14). The missing fuzzy rules for WM method are obtained using Equation 4.6 with different weight assignment strategies, along with weight

normalization. The obtained outcome for WM method after Stage 4 is summarized in Table 4.24. As depicted in Table 4.24 (b), with $w_{j_1,j_2}^{\text{data}}$, for $A_{j_1=2,j_2=2}^{\#}$, $b_{2,2}^{\#} = 0.2467$, $A_{j_1=2,j_2=3}^{\#}$, $b_{2,3}^{\#} = 0.2804$, and $A_{j_1=3,j_2=2}^{\#}$, $b_{3,2}^{\#} = 0.2752$ are obtained, respectively.

Table 4.24: Fuzzy rules for $k_1 = k_2 = 5$ after Stage 4 for triangular fuzzy set using WM method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$					
$A_{1,5}$	0.5000	0.5463	0.7327	0.9263	0.9725
$A_{1,4}$	0.4538	**0.4800**	0.6995	0.8733	0.9463
$A_{1,3}$	0.2610	**0.4948**	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0738	0.4326	**0.4950**	**0.4867**	0.5263
$A_{1,1}$	0.0323	0.0677	0.2610	0.4538	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(a)

With $w_{j_1,j_2}^{\text{data}}$					
$A_{1,5}$	0.5000	0.5463	0.7327	0.9263	0.9725
$A_{1,4}$	0.4538	0.4800	0.6995	0.8733	0.9463
$A_{1,3}$	0.2610	0.2752	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0738	0.2467	0.2804	0.4867	0.5263
$A_{1,1}$	0.0323	0.0677	0.2610	0.4538	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(b)

With w_{j_1,j_2}^{sum}					
$A_{1,5}$	0.5000	0.5463	0.7327	0.9263	0.9725
$A_{1,4}$	0.4538	0.4800	0.6995	0.8733	0.9463
$A_{1,3}$	0.2610	0.3330	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0738	0.2864	0.2914	0.4867	0.5263
$A_{1,1}$	0.0323	0.0677	0.2610	0.4538	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(c)

Table 4.24 continued

	With $w_{j_1,j_2}^{\max}$				
$A_{1,5}$	0.5000	0.5463	0.7327	0.9263	0.9725
$A_{1,4}$	0.4538	0.4800	0.6995	0.8733	0.9463
$A_{1,3}$	0.2610	0.3686	0.5409	**0.7600**	**0.6500**
$A_{1,2}$	0.0738	0.3254	0.3645	0.4867	0.5263
$A_{1,1}$	0.0323	0.0677	0.2610	0.4538	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(d)

Then, for the *complete* and *non-monotone* fuzzy If-Then rules with WM method obtained after Stage 4 using triangular fuzzy set (see Table 4.24) and the *complete* and *non-monotone* fuzzy If-Then rules with Ishibuchi method obtained after Stage 2 using triangular fuzzy set (see Table 4.14) are re-labelled at Stage 5. The re-labelled fuzzy If-Then rules, i.e., $\boldsymbol{b}_j^{\text{rel}}$ **,** using Algorithm 1 with four weight assignment strategies along and weight normalization for WM and Ishibuchi methods using triangular fuzzy set are presented in Table 4.25 and Table 4.26, respectively.

For WM method, two termination thresholds, i.e., $\varepsilon_1 = 10^{-12}$,$\varepsilon_2 = 10^{-12}$, an initial search interval for the direct search method, i.e., $\left[\underline{\beta}, \overline{\beta}\right] = [0.0323, 0.9725]$, and the limit of the central finite difference method, i.e., $\Delta = 0.001$, are used. For the Ishibuchi method, two termination thresholds, i.e., $\varepsilon_1 = 10^{-11}$,$\varepsilon_2 = 10^{-10}$, an initial search interval for the direct search method, i.e., $\left[\underline{\beta}, \overline{\beta}\right] = [0.0385, 0.9667]$, and the limit of the central finite difference method, i.e., $\Delta = 0.001$, are used.

Table 4.25: Fuzzy rules for $k_1 = k_2 = 5$ after Stage 5 for triangular fuzzy set using WM method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

	With $w_{j_1,j_2}^{\text{One-Zero}}$				
$A_{1,5}$	0.5000	0.5463	0.7327	0.9263	0.9725
$A_{1,4}$	0.4538	0.4800	0.6995	0.8733	0.9463
$A_{1,3}$	0.2610	0.4782	0.5409	0.7050	0.7050
$A_{1,2}$	0.0738	0.4326	0.4857	0.4867	0.5263
$A_{1,1}$	0.0323	0.0677	0.2610	0.4538	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(a)

	With $w_{j_1,j_2}^{\text{data}}$				
$A_{1,5}$	0.5000	0.5463	0.7327	0.9263	0.9725
$A_{1,4}$	0.4538	0.4800	0.6995	0.8733	0.9463
$A_{1,3}$	0.2610	0.2752	0.5409	0.7050	0.7050
$A_{1,2}$	0.0738	0.2467	0.2804	0.4867	0.5263
$A_{1,1}$	0.0323	0.0677	0.2610	0.4538	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(b)

	With w_{j_1,j_2}^{sum}				
$A_{1,5}$	0.5000	0.5463	0.7327	0.9263	0.9725
$A_{1,4}$	0.4538	0.4800	0.6995	0.8733	0.9463
$A_{1,3}$	0.2610	0.3330	0.5409	0.6929	0.6929
$A_{1,2}$	0.0738	0.2864	0.2914	0.4867	0.5263
$A_{1,1}$	0.0323	0.0677	0.2610	0.4538	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(c)

Table 4.25 continued

With $w_{j_1,j_2}^{\max}$					
$A_{1,5}$	0.5000	0.5463	0.7327	0.9263	0.9725
$A_{1,4}$	0.4538	0.4800	0.6995	0.8733	0.9463
$A_{1,3}$	0.2610	0.3686	0.5409	0.6929	0.6929
$A_{1,2}$	0.0738	0.3254	0.3645	0.4867	0.5263
$A_{1,1}$	0.0323	0.0677	0.2610	0.4538	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(d)

Table 4.26: Fuzzy rules for $k_1 = k_2 = 5$ after Stage 5 for triangular fuzzy set using Ishibuchi method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$					
$A_{1,5}$	0.5000	0.5449	0.7376	0.9220	0.9667
$A_{1,4}$	0.4448	0.4990	0.7075	0.8578	0.9500
$A_{1,3}$	0.2633	0.3105	0.5878	0.6840	0.6840
$A_{1,2}$	0.0816	0.1130	0.3279	0.4790	0.5375
$A_{1,1}$	0.0385	0.0791	0.2681	0.4479	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(a)

With $w_{j_1,j_2}^{\text{data}}$					
$A_{1,5}$	0.5000	0.5449	0.7376	0.9220	0.9667
$A_{1,4}$	0.4448	0.4990	0.7075	0.8578	0.9500
$A_{1,3}$	0.2633	0.3105	0.5878	0.7179	0.7307
$A_{1,2}$	0.0816	0.1130	0.3279	0.4790	0.5375
$A_{1,1}$	0.0385	0.0791	0.2681	0.4479	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(b)

Table 4.26 continued

With w_{j_1,j_2}^{sum}					
$A_{1,5}$	0.5000	0.5449	0.7376	0.9220	0.9667
$A_{1,4}$	0.4448	0.4990	0.7075	0.8578	0.9500
$A_{1,3}$	0.2633	0.3105	0.5878	0.7179	0.7374
$A_{1,2}$	0.0816	0.1130	0.3279	0.4790	0.5375
$A_{1,1}$	0.0385	0.0791	0.2681	0.4479	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(c)

With w_{j_1,j_2}^{max}					
$A_{1,5}$	0.5000	0.5449	0.7376	0.9220	0.9667
$A_{1,4}$	0.4448	0.4990	0.7075	0.8578	0.9500
$A_{1,3}$	0.2633	0.3105	0.5878	0.6667	0.6667
$A_{1,2}$	0.0816	0.1130	0.3279	0.4790	0.5375
$A_{1,1}$	0.0385	0.0791	0.2681	0.4479	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

(d)

After Stage 5, *complete and monotone* fuzzy If-Then rules are produced for both methods using triangular fuzzy set. The obtained difference scores using WM method for $w_{j_1,j_2}^{\text{One-Zero}}, w_{j_1,j_2}^{\text{data}}, w_{j_1,j_2}^{\text{sum}}, w_{j_1,j_2}^{\text{max}}$ are 0.0060, 1.6812e-04, 1.3727e-04, and 0.0017, respectively. While, the obtained difference scores using Ishibuchi method for $w_{j_1,j_2}^{\text{One-Zero}}, w_{j_1,j_2}^{\text{data}}, w_{j_1,j_2}^{\text{sum}}, w_{j_1,j_2}^{\text{max}}$ are 0.0023, 3.2650e-07, 3.4117e-07, and 1.9343e-04, respectively.

For $k_1 = k_2 = 7$, the weights obtained for the WM and Ishibuchi methods using trapezoidal fuzzy sets after Stage 3 are summarized in Table 4.27 and Table 4.28

respectively. For the WM method, $j_1 = 2$ and $j_2 = 1$, i.e., $b_{2,1} = 0.0541$, the obtained weights are: $w_{2,1}^{\text{One-Zero}} = 1$, $w_{2,1}^{\text{data}} = 2$, $w_{2,1}^{\text{sum}} = 1.6996$ and $w_{2,1}^{\max} = 1$, respectively.

Table 4.27: The values of $\boldsymbol{b_j}$ and weights obtained from WM method for $k_1 = k_2 = 7$ using trapezoidal fuzzy set after Stage 3

j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$	j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$
1	1	0.0326	1.0000	3.0000	1.8887	0.6996	4	5	NaN	0.0000	0.0000	0.0000	0.0000
1	2	0.0541	1.0000	2.0000	1.6996	1.0000	4	6	NaN	0.0000	0.0000	0.0000	0.0000
1	3	0.1346	1.0000	3.0000	2.2591	0.8999	4	7	NaN	0.0000	0.0000	0.0000	0.0000
1	4	0.2641	1.0000	2.0000	1.6996	1.0000	5	1	0.3965	1.0000	4.0000	2.8887	0.8999
1	5	0.3965	1.0000	4.0000	2.8887	0.8999	5	2	0.4600	1.0000	1.0000	0.8999	0.8999
1	6	0.4741	1.0000	2.0000	1.6996	1.0000	5	3	0.5700	1.0000	1.0000	0.7197	0.7197
1	7	0.5000	1.0000	4.0000	2.8887	1.0000	5	4	0.6324	1.0000	2.0000	1.6996	0.8999
2	1	0.0541	1.0000	2.0000	1.6996	1.0000	5	5	0.7641	1.0000	4.0000	2.8887	0.8098
2	2	NaN	0.0000	0.0000	0.0000	0.0000	5	6	NaN	0.0000	0.0000	0.0000	0.0000
2	3	NaN	0.0000	0.0000	0.0000	0.0000	5	7	NaN	0.0000	0.0000	0.0000	0.0000
2	4	NaN	0.0000	0.0000	0.0000	0.0000	6	1	0.4741	1.0000	2.0000	1.6996	1.0000
2	5	0.4600	1.0000	1.0000	0.8999	0.8999	6	2	0.5000	1.0000	1.0000	1.0000	1.0000
2	6	0.5000	1.0000	1.0000	1.0000	1.0000	6	3	NaN	0.0000	0.0000	0.0000	0.0000
2	7	0.5259	1.0000	2.0000	1.6996	1.0000	6	4	0.7100	1.0000	1.0000	1.0000	1.0000
3	1	0.1271	1.0000	4.0000	2.8887	0.8999	6	5	0.8371	1.0000	2.0000	1.6996	0.8999
3	2	NaN	0.0000	0.0000	0.0000	0.0000	6	6	0.9200	1.0000	1.0000	1.0000	1.0000
3	3	0.3100	1.0000	1.0000	0.6396	0.6396	6	7	0.9459	1.0000	2.0000	1.6996	1.0000
3	4	NaN	0.0000	0.0000	0.0000	0.0000	7	1	0.5000	1.0000	3.0000	2.3992	1.0000
3	5	0.5000	1.0000	1.0000	0.8098	0.8098	7	2	0.5259	1.0000	2.0000	1.6996	1.0000
3	6	NaN	0.0000	0.0000	0.0000	0.0000	7	3	0.5857	1.0000	3.0000	2.0889	0.8999
3	7	0.6500	1.0000	1.0000	0.7997	0.7997	7	4	0.7300	1.0000	1.0000	0.6996	0.6996
4	1	0.2641	1.0000	2.0000	1.6996	1.0000	7	5	0.8856	1.0000	3.0000	2.0889	0.8999
4	2	NaN	0.0000	0.0000	0.0000	0.0000	7	6	0.9459	1.0000	2.0000	1.6996	1.0000
4	3	NaN	0.0000	0.0000	0.0000	0.0000	7	7	0.9718	1.0000	4.0000	2.8887	1.0000
4	4	NaN	0.0000	0.0000	0.0000	0.0000							

Table 4.28: The values of $\boldsymbol{b_j}$ and weights obtained from Ishibuchi method for $k_1 = k_2 = 7$ using trapezoidal fuzzy set after Stage 3

j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$	j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$
1	1	0.0326	1.0000	3.0000	1.8887	0.6996	4	5	0.7271	1.0000	2.0000	0.3404	0.1802
1	2	0.0513	1.0000	5.0000	2.3103	1.0000	4	6	0.7600	1.0000	1.0000	0.0201	0.0201
1	3	0.1346	1.0000	3.0000	2.2591	0.8999	4	7	0.6500	1.0000	1.0000	0.2003	0.2003
1	4	0.2627	1.0000	6.0000	2.3803	1.0000	5	1	0.3965	1.0000	4.0000	2.8887	0.8999
1	5	0.3965	1.0000	4.0000	2.8887	0.8999	5	2	0.4391	1.0000	3.0000	1.4104	0.8999
1	6	0.4755	1.0000	6.0000	2.3803	1.0000	5	3	0.5700	1.0000	1.0000	0.7197	0.7197
1	7	0.5000	1.0000	4.0000	2.8887	1.0000	5	4	0.6418	1.0000	5.0000	2.2202	0.8999
2	1	0.0527	1.0000	6.0000	2.3803	1.0000	5	5	0.7641	1.0000	4.0000	2.8887	0.8098
2	2	0.0592	1.0000	4.0000	0.7211	0.3004	5	6	0.7971	1.0000	2.0000	0.1702	0.0901
2	3	0.1700	1.0000	1.0000	0.2402	0.2402	5	7	NaN	0.0000	0.0000	0.0000	0.0000
2	4	0.2686	1.0000	3.0000	0.4207	0.3004	6	1	0.4732	1.0000	5.0000	2.1702	1.0000
2	5	0.4428	1.0000	4.0000	1.5005	0.8999	6	2	0.4977	1.0000	6.0000	1.7611	1.0000
2	6	0.4968	1.0000	7.0000	1.8312	1.0000	6	3	0.5939	1.0000	3.0000	0.5906	0.2703
2	7	0.5213	1.0000	4.0000	2.2102	1.0000	6	4	0.7221	1.0000	8.0000	1.7612	1.0000
3	1	0.1271	1.0000	4.0000	2.8887	0.8999	6	5	0.8406	1.0000	6.0000	2.3803	0.8999
3	2	0.1329	1.0000	2.0000	0.5105	0.2703	6	6	0.9250	1.0000	6.0000	1.8312	1.0000
3	3	0.3100	1.0000	1.0000	0.6396	0.6396	6	7	0.9505	1.0000	4.0000	2.2102	1.0000
3	4	0.3100	1.0000	1.0000	0.1602	0.1602	7	1	0.5000	1.0000	3.0000	2.3992	1.0000
3	5	0.5000	1.0000	1.0000	0.8098	0.8098	7	2	0.5268	1.0000	5.0000	2.1702	1.0000
3	6	0.5000	1.0000	1.0000	0.0901	0.0901	7	3	0.5857	1.0000	3.0000	2.0889	0.8999
3	7	0.6500	1.0000	1.0000	0.7997	0.7997	7	4	0.7314	1.0000	3.0000	0.9798	0.6996
4	1	0.2627	1.0000	6.0000	2.3803	1.0000	7	5	0.8856	1.0000	3.0000	2.0889	0.8999
4	2	0.2686	1.0000	3.0000	0.4207	0.3004	7	6	0.9473	1.0000	7.0000	2.3803	1.0000
4	3	0.3100	1.0000	1.0000	0.1602	0.1602	7	7	0.9718	1.0000	4.0000	2.8887	1.0000
4	4	0.5643	1.0000	3.0000	0.2805	0.2003							

There are 34 fuzzy rules generated from WM method (i.e., Equation 4.3) using trapezoidal fuzzy set (see Table 4.15) with 15 missing fuzzy rules and there are 48 fuzzy rules generated from Ishibuchi method (i.e., Equation 4.4) using trapezoidal fuzzy set (see Table 4.16) with 1 missing fuzzy rules. The missing fuzzy rules for WM and Ishibuchi methods are obtained using Equation 4.6 with different weight assignment strategies, along with weight normalization. The obtained outcome for WM and Ishibuchi methods after Stage 4 is summarized in Table 4.29 and Table 4.30. As depicted in Table 4.30 (b), with $w_{j_1,j_2}^{\text{data}}$, for $A_{j_1=5,j_2=7}^{\#}$, $b_{5,7}^{\#} = 0.2672$ is obtained.

Table 4.29: Fuzzy rules for $k_1 = k_2 = 7$ after Stage 4 for trapezoidal fuzzy set using WM method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$							
$A_{1,7}$	0.5000	0.5259	0.5857	0.7300	0.8856	0.9459	0.9718
$A_{1,6}$	0.4741	0.5000	**0.5613**	0.7100	0.8371	0.9200	0.9459
$A_{1,5}$	0.3965	0.4600	**0.5700**	0.6324	**0.7641**	**0.6321**	0.6335
$A_{1,4}$	0.2641	0.4579	0.4952	0.5321	0.5595	0.5742	**0.5826**
$A_{1,3}$	0.1271	**0.4136**	**0.3100**	0.4852	0.5000	0.5346	**0.6500**
$A_{1,2}$	0.0541	0.3712	**0.4103**	0.4488	0.4600	0.5000	0.5259
$A_{1,1}$	0.0326	0.0541	0.1346	0.2641	0.3965	0.4741	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(a)

Table 4.29 continued

With $w_{j_1,j_2}^{\text{data}}$							
$A_{1,7}$	0.5000	0.5259	0.5857	0.7300	0.8856	0.9459	0.9718
$A_{1,6}$	0.4741	**0.5000**	**0.2740**	0.7100	0.8371	0.9200	0.9459
$A_{1,5}$	0.3965	0.4600	**0.5700**	0.6324	**0.7641**	**0.3342**	**0.3318**
$A_{1,4}$	**0.2641**	**0.2290**	**0.2415**	0.2683	**0.2912**	**0.2942**	**0.2935**
$A_{1,3}$	0.1271	0.2107	**0.3100**	**0.2373**	**0.5000**	**0.2572**	**0.6500**
$A_{1,2}$	0.0541	0.1919	0.2080	**0.2245**	0.4600	**0.5000**	0.5259
$A_{1,1}$	0.0326	0.0541	0.1346	**0.2641**	0.3965	0.4741	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(b)

With w_{j_1,j_2}^{sum}							
$A_{1,7}$	0.5000	0.5259	0.5857	0.7300	0.8856	0.9459	0.9718
$A_{1,6}$	0.4741	**0.5000**	**0.3007**	0.7100	0.8371	0.9200	0.9459
$A_{1,5}$	0.3965	0.4600	**0.5700**	0.6324	**0.7641**	**0.3672**	**0.3660**
$A_{1,4}$	**0.2641**	**0.2507**	**0.2640**	0.2937	**0.3177**	0.3225	**0.3227**
$A_{1,3}$	0.1271	0.2293	**0.3100**	**0.2596**	**0.5000**	**0.2842**	**0.6500**
$A_{1,2}$	0.0541	0.2091	0.2267	**0.2461**	0.4600	0.5000	0.5259
$A_{1,1}$	0.0326	0.0541	0.1346	**0.2641**	0.3965	0.4741	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(c)

With $w_{j_1,j_2}^{\max}$							
$A_{1,7}$	0.5000	0.5259	0.5857	0.7300	0.8856	0.9459	0.9718
$A_{1,6}$	0.4741	0.5000	**0.5103**	0.7100	0.8371	0.9200	0.9459
$A_{1,5}$	0.3965	0.4600	**0.5700**	0.6324	**0.7641**	**0.5825**	0.5904
$A_{1,4}$	0.2641	0.4147	0.4429	0.4787	0.5055	0.5240	**0.5342**
$A_{1,3}$	0.1271	**0.3731**	**0.3100**	0.4360	**0.5000**	**0.4878**	**0.6500**
$A_{1,2}$	0.0541	0.3369	**0.3706**	0.4081	0.4600	**0.5000**	0.5259
$A_{1,1}$	0.0326	0.0541	0.1346	0.2641	0.3965	0.4741	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(d)

Table 4.30: Fuzzy rules for $k_1 = k_2 = 7$ after Stage 4 for trapezoidal fuzzy set using Ishibuchi method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$							
$A_{1,7}$	0.5000	0.5268	**0.5857**	0.7314	0.8856	0.9473	0.9718
$A_{1,6}$	0.4732	0.4977	**0.5939**	0.7221	0.8406	0.9250	0.9505
$A_{1,5}$	0.3965	0.4391	0.5700	0.6418	0.7641	**0.7971**	**0.6094**
$A_{1,4}$	0.2627	0.2686	0.3100	0.5643	0.7271	**0.7600**	**0.6500**
$A_{1,3}$	0.1271	0.1329	0.3100	0.3100	0.5000	0.5000	0.6500
$A_{1,2}$	0.0527	0.0592	0.1700	0.2686	0.4428	0.4968	0.5213
$A_{1,1}$	0.0326	0.0513	0.1346	0.2627	0.3965	0.4755	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(a)

With $w_{j_1,j_2}^{\text{data}}$							
$A_{1,7}$	0.5000	0.5268	**0.5857**	0.7314	0.8856	0.9473	0.9718
$A_{1,6}$	0.4732	0.4977	**0.5939**	0.7221	0.8406	0.9250	0.9505
$A_{1,5}$	0.3965	0.4391	0.5700	0.6418	0.7641	**0.7971**	**0.2672**
$A_{1,4}$	0.2627	0.2686	0.3100	0.5643	0.7271	**0.7600**	**0.6500**
$A_{1,3}$	0.1271	0.1329	0.3100	0.3100	0.5000	0.5000	0.6500
$A_{1,2}$	0.0527	0.0592	0.1700	0.2686	0.4428	0.4968	0.5213
$A_{1,1}$	0.0326	0.0513	0.1346	0.2627	0.3965	0.4755	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(b)

With w_{j_1,j_2}^{sum}							
$A_{1,7}$	0.5000	0.5268	**0.5857**	0.7314	0.8856	0.9473	0.9718
$A_{1,6}$	0.4732	0.4977	**0.5939**	0.7221	0.8406	0.9250	0.9505
$A_{1,5}$	0.3965	0.4391	0.5700	0.6418	0.7641	**0.7971**	**0.3058**
$A_{1,4}$	0.2627	0.2686	0.3100	0.5643	0.7271	**0.7600**	**0.6500**
$A_{1,3}$	0.1271	0.1329	0.3100	0.3100	0.5000	0.5000	0.6500
$A_{1,2}$	0.0527	0.0592	0.1700	0.2686	0.4428	0.4968	0.5213
$A_{1,1}$	0.5000	0.5268	**0.5857**	0.7314	0.8856	0.9473	0.9718
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(c)

Table 4.30 continued

With $w_{j_1,j_2}^{\max}$							
$A_{1,7}$	0.5000	0.5268	**0.5857**	0.7314	0.8856	0.9473	0.9718
$A_{1,6}$	0.4732	0.4977	**0.5939**	0.7221	0.8406	0.9250	0.9505
$A_{1,5}$	0.3965	0.4391	0.5700	0.6418	0.7641	**0.7971**	**0.4089**
$A_{1,4}$	0.2627	0.2686	0.3100	0.5643	0.7271	**0.7600**	**0.6500**
$A_{1,3}$	0.1271	0.1329	0.3100	0.3100	0.5000	0.5000	0.6500
$A_{1,2}$	0.0527	0.0592	0.1700	0.2686	0.4428	0.4968	0.5213
$A_{1,1}$	0.0326	0.0513	0.1346	0.2627	0.3965	0.4755	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(d)

Then, for the c*omplete* and *non-monotone* fuzzy If-Then rules with WM and Ishibuchi methods obtained after Stage 4 using trapezoidal fuzzy set (see Table 4.29 and Table 4.30) are re-labelled at Stage 5, respectively. The re-labelled fuzzy If-Then rules, i.e., $\boldsymbol{b}_j^{\text{rel}}$ **,** using Algorithm 1 with four weight assignment strategies along and weight normalization for WM and Ishibuchi methods are presented in Table 4.31 and Table 4.32, respectively.

For WM and Ishibuchi methods using trapezoidal fuzzy set, two termination thresholds, i.e., $\varepsilon_1 = 10^{-11}$,$\varepsilon_2 = 10^{-10}$, an initial search interval for the direct search method i.e., $\left[\underline{\beta}, \overline{\beta}\right] = [0.0326, 0.9718]$, and the limit of the central finite difference method, i.e., $\Delta = 0.001$, are used.

Table 4.31: Fuzzy rules for $k_1 = k_2 = 7$ after Stage 5 for trapezoidal fuzzy set using WM method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$

$A_{1,7}$	0.5000	0.5259	0.5857	0.7300	0.8856	0.9459	0.9718
$A_{1,6}$	0.4741	0.5000	0.5784	0.7100	0.8371	0.9200	0.9459
$A_{1,5}$	0.3965	0.4600	0.5700	0.6324	0.7641	0.7674	0.7997
$A_{1,4}$	0.2641	0.4579	0.4952	0.5321	0.5595	0.5742	0.7090
$A_{1,3}$	0.1271	0.2865	0.3100	0.4852	0.5000	0.5346	0.6500
$A_{1,2}$	0.0541	0.2732	0.2885	0.4488	0.4600	0.5000	0.5259
$A_{1,1}$	0.0326	0.0541	0.1346	0.2641	0.3965	0.4741	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(a)

With $w_{j_1,j_2}^{\text{data}}$

$A_{1,7}$	0.5000	0.5259	0.5857	0.7300	0.8856	0.9459	0.9718
$A_{1,6}$	0.4740	0.5001	0.5822	0.7100	0.8371	0.9200	0.9459
$A_{1,5}$	0.3965	0.4599	0.5699	0.6325	0.7641	0.7678	0.8189
$A_{1,4}$	0.2641	0.2693	0.3243	0.3451	0.5188	0.5305	0.6513
$A_{1,3}$	0.1271	0.2107	0.3098	0.3123	0.5004	0.5131	0.6500
$A_{1,2}$	0.0541	0.1919	0.2080	0.2936	0.4602	0.5001	0.5259
$A_{1,1}$	0.0326	0.0541	0.1346	0.2641	0.3965	0.4740	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(b)

With w_{j_1,j_2}^{sum}

$A_{1,7}$	0.5000	0.5259	0.5859	0.7300	0.8856	0.9459	0.9718
$A_{1,6}$	0.4739	0.4999	0.5849	0.7100	0.8371	0.9200	0.9459
$A_{1,5}$	0.3964	0.4594	0.5696	0.6321	0.7640	0.7667	0.7960
$A_{1,4}$	0.2641	0.2955	0.3284	0.3367	0.5011	0.5235	0.6511
$A_{1,3}$	0.1271	0.2281	0.3096	0.3297	0.5003	0.5172	0.6500
$A_{1,2}$	0.0541	0.1440	0.2111	0.3032	0.4601	0.5007	0.5255
$A_{1,1}$	0.0326	0.0541	0.1346	0.2641	0.3965	0.4743	0.4999
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(c)

Table 4.31 continued

With $w_{j_1,j_2}^{\max}$							
$A_{1,7}$	0.5000	0.5259	0.5857	0.7300	0.8856	0.9459	0.9718
$A_{1,6}$	0.4741	0.5000	0.5747	0.7100	0.8371	0.9200	0.9459
$A_{1,5}$	0.3965	0.4600	0.5701	0.6324	0.7641	0.7669	0.7866
$A_{1,4}$	0.2641	0.4147	0.4429	0.4787	0.5055	0.5240	0.6851
$A_{1,3}$	0.1271	0.3094	0.3101	0.4360	0.5000	0.5066	0.6500
$A_{1,2}$	0.0541	0.3004	0.3094	0.4081	0.4600	0.5000	0.5259
$A_{1,1}$	0.0326	0.0541	0.1346	0.2641	0.3965	0.4741	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(d)

Table 4.32: Fuzzy rules for $k_1 = k_2 = 7$ after Stage 5 for trapezoidal fuzzy set using Ishibuchi method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$							
$A_{1,7}$	0.5000	0.5268	0.5898	0.7314	0.8856	0.9473	0.9718
$A_{1,6}$	0.4732	0.4977	0.5898	0.7221	0.8406	0.9250	0.9505
$A_{1,5}$	0.3965	0.4391	0.5700	0.6418	0.7641	0.7971	0.8052
$A_{1,4}$	0.2627	0.2686	0.3101	0.5643	0.7124	0.7124	0.7124
$A_{1,3}$	0.1271	0.1329	0.3098	0.3101	0.4999	0.5001	0.6500
$A_{1,2}$	0.0527	0.0592	0.1700	0.2686	0.4428	0.4968	0.5213
$A_{1,1}$	0.0326	0.0513	0.1346	0.2627	0.3965	0.4755	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(a)

Table 4.32 continued

With $w_{j_1,j_2}^{\text{data}}$							
$A_{1,7}$	0.5000	0.5268	0.5825	0.7314	0.8856	0.9473	0.9690
$A_{1,6}$	0.4732	0.4977	0.5750	0.7221	0.8406	0.9250	0.9455
$A_{1,5}$	0.3953	0.4315	0.5679	0.6393	0.7653	0.8017	0.9404
$A_{1,4}$	0.2627	0.2692	0.3710	0.5603	0.6861	0.7300	0.7696
$A_{1,3}$	0.1271	0.1329	0.2153	0.3850	0.5190	0.5223	0.7689
$A_{1,2}$	0.0527	0.0592	0.1700	0.2692	0.4427	0.4912	0.5216
$A_{1,1}$	0.0326	0.0513	0.1346	0.2627	0.3953	0.4753	0.4997
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(b)

With w_{j_1,j_2}^{sum}							
$A_{1,7}$	0.5000	0.5268	0.5864	0.7314	0.8856	0.9473	0.9718
$A_{1,6}$	0.4732	0.4977	0.5836	0.7221	0.8406	0.9249	0.9505
$A_{1,5}$	0.3965	0.4391	0.5700	0.6417	0.7640	0.8025	0.8855
$A_{1,4}$	0.2628	0.2716	0.3223	0.5644	0.7068	0.7073	0.7086
$A_{1,3}$	0.1271	0.1329	0.3096	0.3172	0.5015	0.5253	0.6483
$A_{1,2}$	0.0526	0.0592	0.1697	0.2685	0.4438	0.4968	0.5213
$A_{1,1}$	0.0329	0.0513	0.1346	0.2627	0.3965	0.4755	0.4999
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(c)

With $w_{j_1,j_2}^{\max}$							
$A_{1,7}$	0.5000	0.5268	0.5864	0.7314	0.8856	0.9473	0.9718
$A_{1,6}$	0.4732	0.4977	0.5864	0.7221	0.8406	0.9250	0.9505
$A_{1,5}$	0.3965	0.4391	0.5700	0.6418	0.7641	0.7965	0.8531
$A_{1,4}$	0.2627	0.2686	0.3107	0.5643	0.6850	0.6850	0.6852
$A_{1,3}$	0.1271	0.1329	0.3100	0.3107	0.5000	0.5010	0.6500
$A_{1,2}$	0.0527	0.0592	0.1700	0.2686	0.4428	0.4968	0.5213
$A_{1,1}$	0.0326	0.0513	0.1346	0.2627	0.3965	0.4755	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(d)

After Stage 5, *complete and monotone* fuzzy If-Then rules are produced for both methods using trapezoidal fuzzy set. The obtained difference scores using WM method for $w_{j_1,j_2}^{\text{One-Zero}}$, $w_{j_1,j_2}^{\text{data}}$, w_{j_1,j_2}^{sum}, $w_{j_1,j_2}^{\max}$ are 1.0000e-06, 1.0250e-06, 1.2858e-06, and 1.0093e-06, respectively. While, the obtained difference scores using Ishibuchi method for $w_{j_1,j_2}^{\text{One-Zero}}$, $w_{j_1,j_2}^{\text{data}}$, w_{j_1,j_2}^{sum}, $w_{j_1,j_2}^{\max}$ are 1.0000e-06, 1.0250e-06, 1.2858e-06, and 1.0093e-06, respectively.

For $k_1 = k_2 = 7$, the weights obtained for the WM and Ishibuchi methods using triangular fuzzy sets after Stage 3 are summarized in Table 4.33 and Table 4.34 respectively. For the WM method, $j_1 = 3$ and $j_2 = 3$, i.e., $b_{3,3} = 0.3100$, the obtained weights are: $w_{3,3}^{\text{One-Zero}} = 1$, $w_{3,3}^{\text{data}} = 1$, $w_{3,3}^{\text{sum}} = 0.3586$ and $w_{3,3}^{\max} = 0.3586$, respectively. For $j_1 = 3$ and $j_2 = 4$, $b_{3,4}$ is *unknown*, resulting in $w_{3,4}^{\text{One-Zero}} = 0$, $w_{3,4}^{\text{data}} = 0$, $w_{3,4}^{\text{sum}} = 0$ and $w_{3,4}^{\max} = 0$, respectively.

Table 4.33: The values of $\boldsymbol{b_j}$ and weights obtained from WM method for $k_1 = k_2 = 7$ using triangular fuzzy set after Stage 3

j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$	j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$
1	1	NaN	0.0000	0.0000	0.0000	0.0000	4	5	NaN	0.0000	0.0000	0.0000	0.0000
1	2	0.0414	1.0000	2.0000	1.3988	0.7964	4	6	NaN	0.0000	0.0000	0.0000	0.0000
1	3	0.1199	1.0000	2.0000	1.4012	0.8024	4	7	NaN	0.0000	0.0000	0.0000	0.0000
1	4	0.2600	1.0000	1.0000	1.0000	1.0000	5	1	0.3958	1.0000	2.0000	1.4012	0.8024
1	5	0.3958	1.0000	2.0000	1.4012	0.8024	5	2	0.4292	1.0000	3.0000	1.4831	0.6390
1	6	0.4786	1.0000	2.0000	1.3988	0.7964	5	3	0.5700	1.0000	1.0000	0.4805	0.4805
1	7	0.5000	1.0000	1.0000	1.0000	1.0000	5	4	0.6358	1.0000	2.0000	1.4012	0.8024
2	1	0.0414	1.0000	2.0000	1.3988	0.7964	5	5	0.7702	1.0000	4.0000	1.9634	0.6438
2	2	0.0545	1.0000	3.0000	1.3224	0.4798	5	6	NaN	0.0000	0.0000	0.0000	0.0000
2	3	0.1700	1.0000	1.0000	0.3607	0.3607	5	7	NaN	0.0000	0.0000	0.0000	0.0000
2	4	0.2700	1.0000	1.0000	0.6024	0.6024	6	1	0.4786	1.0000	2.0000	1.3988	0.7964
2	5	0.4292	1.0000	3.0000	1.4831	0.6390	6	2	0.5000	1.0000	3.0000	1.5938	0.6343
2	6	0.5000	1.0000	4.0000	1.9567	0.6343	6	3	0.5942	1.0000	2.0000	0.8441	0.4834
2	7	0.5214	1.0000	2.0000	1.3988	0.7964	6	4	0.7186	1.0000	2.0000	1.3988	0.7964
3	1	0.1199	1.0000	2.0000	1.4012	0.8024	6	5	0.8530	1.0000	4.0000	1.9600	0.6390
3	2	0.1299	1.0000	2.0000	0.8441	0.4834	6	6	0.9372	1.0000	4.0000	1.9567	0.6343
3	3	0.3100	1.0000	1.0000	0.3586	0.3586	6	7	0.9586	1.0000	2.0000	1.3988	0.7964
3	4	NaN	0.0000	0.0000	0.0000	0.0000	7	1	0.5000	1.0000	1.0000	1.0000	1.0000
3	5	0.5000	1.0000	1.0000	0.6438	0.6438	7	2	0.5214	1.0000	2.0000	1.3988	0.7964
3	6	NaN	0.0000	0.0000	0.0000	0.0000	7	3	0.5700	1.0000	1.0000	0.8024	0.8024
3	7	0.6500	1.0000	1.0000	0.5988	0.5988	7	4	NaN	0.0000	0.0000	0.0000	0.0000
4	1	0.2600	1.0000	1.0000	1.0000	1.0000	7	5	0.9100	1.0000	1.0000	0.8024	0.8024
4	2	0.2700	1.0000	1.0000	0.6024	0.6024	7	6	0.9586	1.0000	2.0000	1.3988	0.7964
4	3	NaN	0.0000	0.0000	0.0000	0.0000	7	7	0.9800	1.0000	1.0000	1.0000	1.0000
4	4	NaN	0.0000	0.0000	0.0000	0.0000							

Table 4.34: The values of $\boldsymbol{b_j}$ and weights obtained from Ishibuchi method for $k_1 = k_2 = 7$ using triangular fuzzy set after Stage 3

j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$	j_1	j_2	b_{j_1,j_2}	$w_{j_1,j_2}^{\text{One-Zero}}$	$w_{j_1,j_2}^{\text{data}}$	w_{j_1,j_2}^{sum}	$w_{j_1,j_2}^{\max}$
1	1	0.0317	1.0000	3.0000	0.9533	0.3976	4	5	0.7301	1.0000	2.0000	0.5622	0.3219
1	2	0.0484	1.0000	5.0000	2.1526	0.7964	4	6	0.7600	1.0000	1.0000	0.0793	0.0793
1	3	0.1162	1.0000	5.0000	1.9238	0.8024	4	7	0.6500	1.0000	1.0000	0.4012	0.4012
1	4	0.2606	1.0000	6.0000	2.5190	1.0000	5	1	0.4081	1.0000	6.0000	2.2428	0.8024
1	5	0.4081	1.0000	6.0000	2.2428	0.8024	5	2	0.4392	1.0000	5.0000	1.7679	0.6390
1	6	0.4754	1.0000	6.0000	2.2311	0.7964	5	3	0.5395	1.0000	3.0000	0.6853	0.4805
1	7	0.5000	1.0000	4.0000	1.9533	1.0000	5	4	0.6593	1.0000	7.0000	2.5705	0.8024
2	1	0.0502	1.0000	6.0000	2.2311	0.7964	5	5	0.7816	1.0000	7.0000	2.2901	0.6438
2	2	0.0583	1.0000	4.0000	1.4415	0.4798	5	6	0.8656	1.0000	5.0000	0.6019	0.1621
2	3	0.1421	1.0000	2.0000	0.4834	0.3607	5	7	0.9472	1.0000	2.0000	0.2845	0.2036
2	4	0.2678	1.0000	3.0000	1.0858	0.6024	6	1	0.4725	1.0000	5.0000	1.9916	0.7964
2	5	0.4442	1.0000	6.0000	1.9265	0.6390	6	2	0.4968	1.0000	6.0000	1.9892	0.6343
2	6	0.4941	1.0000	7.0000	2.2721	0.6343	6	3	0.5688	1.0000	6.0000	1.2874	0.4834
2	7	0.5185	1.0000	4.0000	1.9550	0.7964	6	4	0.7248	1.0000	8.0000	2.5579	0.7964
3	1	0.1139	1.0000	6.0000	2.2428	0.8024	6	5	0.8557	1.0000	8.0000	2.5217	0.6390
3	2	0.1210	1.0000	3.0000	0.9667	0.4834	6	6	0.9288	1.0000	7.0000	2.2721	0.6343
3	3	0.3100	1.0000	1.0000	0.3586	0.3586	6	7	0.9558	1.0000	4.0000	1.9550	0.7964
3	4	0.3100	1.0000	1.0000	0.2402	0.2402	7	1	0.5000	1.0000	3.0000	1.7952	1.0000
3	5	0.4923	1.0000	3.0000	0.8487	0.6438	7	2	0.5275	1.0000	5.0000	1.9916	0.7964
3	6	0.5017	1.0000	4.0000	0.4836	0.1621	7	3	0.5708	1.0000	5.0000	1.6440	0.8024
3	7	0.6104	1.0000	3.0000	0.8833	0.5988	7	4	0.7322	1.0000	3.0000	0.7166	0.3976
4	1	0.2606	1.0000	6.0000	2.5190	1.0000	7	5	0.9029	1.0000	5.0000	1.6440	0.8024
4	2	0.2678	1.0000	3.0000	1.0858	0.6024	7	6	0.9498	1.0000	6.0000	2.2311	0.7964
4	3	0.3100	1.0000	1.0000	0.2402	0.2402	7	7	0.9743	1.0000	4.0000	1.9533	1.0000
4	4	0.5499	1.0000	3.0000	0.7231	0.4012							

There are 38 fuzzy rules generated from WM method (i.e., Equation 4.3) using triangular fuzzy set (see Table 4.17) with 11 missing fuzzy rules and there are 49 fuzzy rules generated from Ishibuchi method (i.e., Equation 4.34) using triangular fuzzy set (see Table 4.18). The missing fuzzy rules for WM method are obtained using Equation 4.6 with different weight assignment strategies, along with weight normalization. The obtained outcome for WM method after Stage 4 is summarized in Table 4.35.

Table 4.35: Fuzzy rules for $k_1 = k_2 = 7$ after Stage 4 for triangular fuzzy set using WM method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$

$A_{1,7}$	0.5000	0.5214	**0.5700**	**0.5845**	0.9100	0.9586	0.9800
$A_{1,6}$	0.4786	0.5000	**0.5942**	**0.7186**	0.8530	0.9372	0.9586
$A_{1,5}$	0.3958	0.4292	0.5700	0.6358	**0.7702**	**0.5988**	0.6006
$A_{1,4}$	0.2600	0.2700	0.4417	0.4861	0.5196	0.5381	**0.5491**
$A_{1,3}$	0.1199	0.1299	0.3100	0.4318	0.5000	0.5006	**0.6500**
$A_{1,2}$	**0.0414**	0.0545	0.1700	0.2700	0.4292	0.5000	0.5214
$A_{1,1}$	**0.3308**	**0.0414**	0.1199	0.2600	0.3958	0.4786	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(a)

With $w_{j_1,j_2}^{\text{data}}$

$A_{1,7}$	0.5000	0.5214	**0.5700**	**0.3014**	0.9100	0.9586	0.9800
$A_{1,6}$	0.4786	0.5000	**0.5942**	**0.7186**	0.8530	0.9372	0.9586
$A_{1,5}$	0.3958	0.4292	0.5700	0.6358	**0.7702**	**0.3568**	**0.3315**
$A_{1,4}$	0.2600	**0.2700**	**0.2203**	0.2563	**0.2878**	**0.2948**	**0.2887**
$A_{1,3}$	0.1199	0.1299	**0.3100**	**0.2161**	**0.5000**	**0.2629**	**0.6500**
$A_{1,2}$	**0.0414**	0.0545	0.1700	**0.2700**	0.4292	**0.5000**	0.5214
$A_{1,1}$	**0.1648**	**0.0414**	0.1199	0.2600	0.3958	0.4786	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(b)

Table 4.35 continued

With w_{j_1,j_2}^{sum}

$A_{1,7}$	0.5000	0.5214	**0.5700**	**0.3727**	0.9100	0.9586	0.9800
$A_{1,6}$	0.4786	0.5000	**0.5942**	**0.7186**	0.8530	0.9372	0.9586
$A_{1,5}$	0.3958	0.4292	0.5700	0.6358	**0.7702**	**0.4209**	**0.4062**
$A_{1,4}$	0.2600	**0.2700**	**0.2640**	0.3073	**0.3416**	**0.3534**	**0.3520**
$A_{1,3}$	0.1199	0.1299	**0.3100**	**0.2595**	**0.5000**	**0.3168**	**0.6500**
$A_{1,2}$	**0.0414**	0.0545	0.1700	**0.2700**	0.4292	**0.5000**	0.5214
$A_{1,1}$	**0.2032**	**0.0414**	0.1199	0.2600	0.3958	0.4786	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(c)

With $w_{j_1,j_2}^{\max}$

$A_{1,7}$	0.5000	0.5214	**0.5700**	**0.4288**	0.9100	0.9586	0.9800
$A_{1,6}$	0.4786	0.5000	**0.5942**	**0.7186**	0.8530	0.9372	0.9586
$A_{1,5}$	0.3958	0.4292	0.5700	0.6358	**0.7702**	**0.4319**	0.4429
$A_{1,4}$	0.2600	0.2700	**0.3051**	0.3421	**0.3685**	0.3856	**0.3968**
$A_{1,3}$	0.1199	0.1299	**0.3100**	**0.3004**	**0.5000**	**0.3564**	**0.6500**
$A_{1,2}$	**0.0414**	0.0545	0.1700	0.2700	0.4292	**0.5000**	0.5214
$A_{1,1}$	**0.2373**	**0.0414**	0.1199	0.2600	0.3958	0.4786	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(d)

Then, for the *complete* and *non-monotone* fuzzy If-Then rules with WM method obtained after Stage 4 using triangular fuzzy set (see Table 4.35) and *complete* and *non-monotone* fuzzy If-Then rules for $k_1 = k_2 = 7$ with the Ishibuchi method obtained after Stage 2 using triangular fuzzy set (see Table 4.18) are re-labelled at Stage 5, respectively. The re-labelled fuzzy If-Then rules, i.e., $\boldsymbol{b}_j^{\text{rel}}$**,** using Algorithm 1 with four weight assignment strategies along and weight normalization for WM and Ishibuchi methods are presented in Table 4.36 and Table 4.37, respectively.

For WM and Ishibuchi methods, two termination thresholds, i.e., $\varepsilon_1 = 10^{-11}$,$\varepsilon_2 = 10^{-10}$, an initial search interval for the direct search method i.e., $\left[\underline{\beta}, \overline{\beta}\right] = [0.0414, 0.9800]$ and $\left[\underline{\beta}, \overline{\beta}\right] = [0.0317, 0.9743]$, respectively, and the limit of the central finite difference method, i.e., $\Delta = 0.001$, are used.

Table 4.36: Fuzzy rules for $k_1 = k_2 = 7$ after Stage 5 for triangular fuzzy set using WM method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$							
$A_{1,7}$	0.5000	0.5214	0.5821	0.7876	0.9100	0.9586	0.9800
$A_{1,6}$	0.4786	0.5000	0.5821	0.7186	0.8530	0.9372	0.9586
$A_{1,5}$	0.3958	0.4292	0.5700	0.6358	0.7702	0.7803	0.7917
$A_{1,4}$	0.2600	0.2700	0.4417	0.4861	0.5196	0.5381	0.6999
$A_{1,3}$	0.1199	0.1299	0.3100	0.4318	0.5000	0.5009	0.6500
$A_{1,2}$	0.0414	0.0545	0.1700	0.2700	0.4292	0.5000	0.5214
$A_{1,1}$	-0.0651	0.0414	0.1199	0.2600	0.3958	0.4786	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(a)

With $w_{j_1,j_2}^{\text{data}}$							
$A_{1,7}$	0.5000	0.5214	0.5894	0.7202	0.9100	0.9586	0.9800
$A_{1,6}$	0.4786	0.5000	0.5894	0.7186	0.8530	0.9372	0.9586
$A_{1,5}$	0.3958	0.4292	0.5700	0.6358	0.7702	0.7766	0.7848
$A_{1,4}$	0.2600	0.2700	0.3123	0.3914	0.5022	0.5219	0.6593
$A_{1,3}$	0.1199	0.1299	0.3100	0.3118	0.5000	0.5079	0.6500
$A_{1,2}$	0.0414	0.0545	0.1700	0.2700	0.4292	0.5000	0.5214
$A_{1,1}$	-0.1220	0.0414	0.1199	0.2600	0.3958	0.4786	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(b)

Table 4.36 continued

With w_{j_1,j_2}^{sum}							
$A_{1,7}$	0.5000	0.5214	0.5827	0.7304	0.9100	0.9586	0.9800
$A_{1,6}$	0.4786	0.5000	0.5827	0.7186	0.8530	0.9372	0.9586
$A_{1,5}$	0.3958	0.4292	0.5700	0.6358	0.7702	0.7785	0.7807
$A_{1,4}$	0.2600	0.2700	0.3123	0.3694	0.5076	0.5126	0.6590
$A_{1,3}$	0.1199	0.1299	0.3100	0.3109	0.5000	0.5064	0.6500
$A_{1,2}$	0.0414	0.0545	0.1700	0.2700	0.4292	0.5000	0.5214
$A_{1,1}$	-0.0017	0.0414	0.1199	0.2600	0.3958	0.4786	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(c)

With $w_{j_1,j_2}^{\max}$							
$A_{1,7}$	0.5000	0.5214	0.5764	0.7236	0.9100	0.9586	0.9800
$A_{1,6}$	0.4786	0.5000	0.5764	0.7186	0.8530	0.9372	0.9586
$A_{1,5}$	0.3958	0.4292	0.5700	0.6358	0.7702	0.7729	0.7738
$A_{1,4}$	0.2600	0.2700	0.3115	0.3421	0.5039	0.5188	0.6568
$A_{1,3}$	0.1199	0.1299	0.3100	0.3196	0.5000	0.5127	0.6500
$A_{1,2}$	0.0414	0.0545	0.1700	0.2700	0.4292	0.5000	0.5214
$A_{1,1}$	0.0400	0.0414	0.1199	0.2600	0.3958	0.4786	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(d)

Table 4.37: Fuzzy rules for $k_1 = k_2 = 7$ after Stage 5 for triangular fuzzy set using Ishibuchi method (a) With $w_{j_1,j_2}^{\text{One-Zero}}$, (b) With $w_{j_1,j_2}^{\text{data}}$, (c) With w_{j_1,j_2}^{sum} and (d) With $w_{j_1,j_2}^{\max}$

With $w_{j_1,j_2}^{\text{One-Zero}}$							
$A_{1,7}$	0.5000	0.5275	0.5708	0.7322	0.9029	0.9498	0.9743
$A_{1,6}$	0.4725	0.4968	0.5688	0.7248	0.8557	0.9288	0.9558
$A_{1,5}$	0.4081	0.4392	0.5395	0.6593	0.7816	0.8656	0.9472
$A_{1,4}$	0.2606	0.2678	0.3101	0.5499	0.7134	0.7134	0.7134
$A_{1,3}$	0.1139	0.1210	0.3098	0.3101	0.4923	0.5017	0.6104
$A_{1,2}$	0.0502	0.0583	0.1421	0.2678	0.4442	0.4941	0.5185
$A_{1,1}$	0.0317	0.0484	0.1162	0.2606	0.4081	0.4754	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(a)

With $w_{j_1,j_2}^{\text{data}}$							
$A_{1,7}$	0.5000	0.5275	0.5708	0.7322	0.9029	0.9498	0.9743
$A_{1,6}$	0.4725	0.4968	0.5688	0.7248	0.8557	0.9288	0.9558
$A_{1,5}$	0.4081	0.4392	0.5395	0.6593	0.7816	0.8656	0.9472
$A_{1,4}$	0.2606	0.2678	0.3115	0.5499	0.7301	0.7333	0.7351
$A_{1,3}$	0.1139	0.1210	0.3070	0.3115	0.4923	0.5017	0.6104
$A_{1,2}$	0.0502	0.0583	0.1421	0.2678	0.4442	0.4941	0.5185
$A_{1,1}$	0.0317	0.0484	0.1162	0.2606	0.4081	0.4754	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(b)

With w_{j_1,j_2}^{sum}							
$A_{1,7}$	0.5000	0.5275	0.5708	0.7322	0.9029	0.9498	0.9743
$A_{1,6}$	0.4725	0.4968	0.5688	0.7248	0.8557	0.9288	0.9558
$A_{1,5}$	0.4081	0.4392	0.5395	0.6593	0.7816	0.8656	0.9472
$A_{1,4}$	0.2612	0.2685	0.3090	0.5500	0.6957	0.6957	0.6959
$A_{1,3}$	0.1131	0.1248	0.3032	0.3116	0.4923	0.5017	0.6104
$A_{1,2}$	0.0502	0.0583	0.1444	0.2675	0.4442	0.4941	0.5185
$A_{1,1}$	0.0317	0.0484	0.1162	0.2594	0.4081	0.4754	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(c)

Table 4.37 continued

	With $w_{j_1,j_2}^{\max}$						
$A_{1,7}$	0.5000	0.5275	0.5708	0.7322	0.9029	0.9498	0.9743
$A_{1,6}$	0.4725	0.4968	0.5688	0.7248	0.8557	0.9288	0.9558
$A_{1,5}$	0.4081	0.4392	0.5395	0.6593	0.7816	0.8656	0.9472
$A_{1,4}$	0.2606	0.2678	0.3104	0.5499	0.6798	0.6798	0.6798
$A_{1,3}$	0.1139	0.1210	0.3097	0.3104	0.4923	0.5017	0.6104
$A_{1,2}$	0.0502	0.0583	0.1421	0.2678	0.4442	0.4941	0.5185
$A_{1,1}$	0.0317	0.0484	0.1162	0.2606	0.4081	0.4754	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

(d)

After Stage 5, c*omplete and monotone* fuzzy If-Then rules are produced for both methods using triangular fuzzy set. The obtained difference scores using WM method for $w_{j_1,j_2}^{\text{One-Zero}}$, $w_{j_1,j_2}^{\text{data}}$, w_{j_1,j_2}^{sum}, $w_{j_1,j_2}^{\max}$ are 2.9393e-04, 2.9342e-05, 5.1629e-05, and 1.0140e-04, respectively. While, the obtained difference scores using Ishibuchi method for $w_{j_1,j_2}^{\text{One-Zero}}$, $w_{j_1,j_2}^{\text{data}}$, w_{j_1,j_2}^{sum}, $w_{j_1,j_2}^{\max}$ are 2.9393e-04, 2.9342e-05, 5.1629e-05, and 1.0140e-04, respectively.

4.4.2 Results and Discussion

The DOM scores of the resulting FIS models are measured using Equation 3.5 (see Chapter 3 Section 3.4), in which $N_1 = N_2 = 20$ is used. The results for the WM and Ishibuchi methods using trapezoidal fuzzy set are depicted in Table 4.38 and Table 4.39, respectively. In Table 4.38 (a), for $k_1 = k_2 = 3$ (see Figure 4.10), the DOM score equals to 1.0000. The SSE value for the training and validation sets is 0.2975 and 0.4212, respectively. On the other hand, as depicted in Table 4.38 (b), for $k_1 = k_2 = 5$ (see Figure 4.12) and Table 4.38 (c), for $k_1 = k_2 = 7$ (see Figure 4.14), columns "*Without relabeling*"

and "*With relabeling*" represent the SSE and DOM scores for the FIS models, based on the fuzzy rules obtained after Stages 4 and 5, respectively. Column "*Strategies*" represents the four weight assignment strategies with Equation 4.5 in Stage 3, based on weight normalization. As an example, in Table 4.38 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.9455 and 1.0000, the SSE values with the training data set are 0.0240 and 0.0206, the SSE values with the validation data set are 0.1133 and 0.0778 for Stages 4 and 5, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled in Stage 5. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing Stage 5.

Table 4.38: Experiment result for WM method using trapezoidal fuzzy set (a) For $k_1 = k_2 = 3$ (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$		
DOM	SSE with training set	SSE with validation set
1.0000	0.2975	0.4212

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without relabeling			With relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.9386	0.0582	0.6336	1.0000	0.0521	0.5628
2	$w_{j_1,j_2}^{\text{data}}$	0.9455	0.0240	0.1133	1.0000	0.0206	0.0778
3	w_{j_1,j_2}^{sum}	0.9659	0.0254	0.1195	1.0000	0.0218	0.0849
4	$w_{j_1,j_2}^{\max}$	0.9659	0.0458	0.4077	1.0000	0.0422	0.3721

(b)

Table 4.38 continued

For $k_1 = k_2 = 7$							
	Strategies	Without relabeling			With relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.9000	0.0673	0.9319	1.0000	0.0428	0.4067
2	$w_{j_1,j_2}^{\text{data}}$	0.7205	0.0638	2.8446	1.0000	0.0218	0.2508
3	w_{j_1,j_2}^{sum}	0.7614	0.0585	2.4754	1.0000	0.0211	0.2700
4	w_{j_1,j_2}^{max}	0.8591	0.0564	0.9728	1.0000	0.0420	0.4280

(c)

Similar results are obtained for the Ishibuchi method using trapezoidal fuzzy set. For $k_1 = k_2 = 3$, *complete* and *monotone* (i.e.,DOM $= 1$) fuzzy If-Then rules are obtained in Stage 2 (see Table 4.8). However, for $k_1 = k_2 = 5$, *complete* and *non-monotone* fuzzy If-Then rules are obtained in Stage 2 (see Table 4.11), therefore Stage 4 (i.e., WFRI) is omitted as the is no missing fuzzy rule. For $k_1 = k_2 = 7$, *incomplete* and *non-monotone* fuzzy If-Then rules are obtained in Stage 2 (see Table 4.16). As an example, in Table 4.39 (c), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.9182 and 1.0000, the SSE values with the training data set are 0.0178 and 0.0327, the SSE values with the validation data set are 0.9706 and 0.1011 for Stages 4 and 5, respectively. Again, the SSE values for both training and validation data set are better (i.e., lower) with the addition of Stages 3 to 5, as compared with those after Stage 2.

Table 4.39: Experiment result for Ishibuchi method using trapezoidal fuzzy set (a) For $k_1 = k_2 = 3$ (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$		
DOM	SSE with training set	SSE with validation set
1.0000	0.2975	0.4212

(a)

For $k_1 = k_2 = 5$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without re-labeling	0.9659	0.0344	0.1607
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	0.0333	0.1348
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	0.0335	0.1179
4	w_{j_1,j_2}^{sum}	1.0000	0.0328	0.1270
5	w_{j_1,j_2}^{max}	1.0000	0.0333	0.1343

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without relabeling			With relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.9182	0.0178	0.2814	1.0000	0.0179	0.0972
2	$w_{j_1,j_2}^{\text{data}}$	0.9182	0.0178	0.9706	1.0000	0.0327	0.1011
3	w_{j_1,j_2}^{sum}	0.9182	0.0178	0.8675	1.0000	0.0177	0.0721
4	w_{j_1,j_2}^{max}	0.9182	0.0178	0.6239	1.0000	0.0175	0.0840

(c)

The results for the WM and Ishibuchi methods using triangular fuzzy set are depicted in Table 4.40 and Table 4.41, respectively. In Table 4.40 (a), for $k_1 = k_2 = 3$ (see Figure 4.11), the DOM score equals to 1.0000. The SSE value for the training and

validation sets is 0.3994 and 0.5692, respectively. As an example, in Table 4.40 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.9273 and 1.0000, the SSE values with the training data set are 0.0386 and 0.0378, the SSE values with the validation data set are 0.0884 and 0.0698 for Stages 4 and 5, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled in Stage 5. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing Stage 5.

Table 4.40: Experiment result for WM method using triangular fuzzy set (a) For $k_1 = k_2 = 3$ (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$		
DOM	SSE with training set	SSE with validation set
1.0000	0.3994	0.5692

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without relabeling			With relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.9045	0.1602	0.6803	1.0000	0.1521	0.6218
2	$w_{j_1,j_2}^{\text{data}}$	0.9273	0.0386	0.0884	1.0000	0.0378	0.0698
3	w_{j_1,j_2}^{sum}	0.9273	0.0514	0.1382	1.0000	0.0514	0.1234
4	$w_{j_1,j_2}^{\max}$	0.9273	0.0721	0.2349	1.0000	0.0719	0.2187

(b)

Table 4.40 continued

For $k_1 = k_2 = 7$							
	Strategies	Without relabeling			With relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8477	0.0904	0.6568	1.0000	0.0332	0.2152
2	$w_{j_1,j_2}^{\text{data}}$	0.6295	0.2254	2.7531	1.0000	0.0361	0.2666
3	w_{j_1,j_2}^{sum}	0.6523	0.1733	2.0284	1.0000	0.0307	0.2625
4	$w_{j_1,j_2}^{\max}$	0.7250	0.1510	1.6843	1.0000	0.0337	0.2783

(c)

Similar results are obtained using the Ishibuchi method. For $k_1 = k_2 = 3$, *complete* and *monotone* (i.e.,DOM $= 1$) fuzzy If-Then rules are obtained in Stage 2 (see Table 4.10). However, for $k_1 = k_2 = 5$ and $k_1 = k_2 = 7$, *complete* and *non-monotone* fuzzy If-Then rules are obtained in Stage 2, see Table 4.14 and Table 4.18, respectively. With this data set, Stage 4 (i.e., WFRI) is omitted as the is no missing fuzzy rule. The DOM scores for $k_1 = k_2 = 5$ and $k_1 = k_2 = 7$ are 0.9432 and 0.9773, respectively, based on the fuzzy If-Then rules obtained from Stage 2. But, the DOM scores for both settings are 1, after Stage 5 (i.e., MFRR), for all four weight assignment strategies. Again, the SSE values for both training and validation data set are better (i.e., lower) with the addition of Stages 3 to 5, as compared with those after Stage 2.

Table 4.41: Experiment result for Ishibuchi method using triangular fuzzy set (a) For $k_1 = k_2 = 3$ (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$		
DOM	SSE with training set	SSE with validation set
1.0000	0.4571	0.4605

(a)

For $k_1 = k_2 = 5$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without relabeling	0.9432	0.0369	0.0629
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	0.0368	0.0527
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	0.0349	0.0429
4	w_{j_1,j_2}^{sum}	1.0000	0.0349	0.0425
5	$w_{j_1,j_2}^{\max}$	1.0000	0.0383	0.0624

(b)

For $k_1 = k_2 = 7$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without relabeling	0.9773	0.0093	0.0431
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	0.0085	0.0302
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	0.0087	0.0321
4	w_{j_1,j_2}^{sum}	1.0000	0.0083	0.0302
5	$w_{j_1,j_2}^{\max}$	1.0000	0.0086	0.0346

(c)

4.5 Summary

In this chapter, *monotone data sets* are defined and a number of simulated examples are introduced. Analyses of the *ad hoc* methods for single and multi-attribute *monotone* data sets are presented. A six-stage framework for practical modeling of *monotone* TSK

FIS models with *ad hoc* methods is proposed. Experiment results are conducted to demonstrate the usefulness of our proposed approach.

In the next chapter, three benchmark problems are used for evaluation to demonstrate the usefulness of our proposed framework for modeling *monotone* TSK FIS models with *ad hoc* methods. The results are analysed and discussed thoroughly.

CHAPTER 5

APPLICATION OF THE SIX-STAGE FRAMEWORK TO PREDICT THE THERMAL COMFORT INDEX AND EVALUATION OF BECHMARK EXAMPLES

5.1 Chapter Overview

In the previous chapter, *monotone data sets* were defined. Analyses of the *ad hoc* methods for single and multi-attribute *monotone* data sets were also presented. A six-stage framework for practical modeling of *monotone* TSK FIS models with *ad hoc* methods was proposed. In this chapter, an application of the six-stage framework to predict the thermal comfort index (Fanger, 1970) (a *monotone* data set) is reported. In addition, two real-world benchmark problems, i.e., the auto-miles per gallon (Auto MPG) problem from the UCI Machine Learning Repository (Asuncion & Newman, 2007) and the mercury-in-fish problem from International Council for the Exploration of the Sea, are used for experimentations too.

In Section 5.2, the background of the benchmark data sets are introduced. In Section 5.3, a series of experiments using benchmark problems are conducted with the six-stage framework. Finally, concluding remarks are given in Section 5.4.

5.2 Data Sets

5.2.1 Benchmark Example 1: Thermal Comfort Heat Index Prediction

An application to prediction of the thermal comfort index, i.e., Predicted Mean Vote (PMV) index as proposed by Fanger (Fanger, 1970), is considered. The PMV index is a function of six variables, namely air temperature, radiant temperature, relative

humidity, air velocity, human activity level, and clothing thermal resistance. The PMV index has a *monotone* relationship with only two attributes, namely air temperature and relative humidity. When the air temperature and relative humidity increase, the PMV index should increase too.

Modelling of the PMV index using a *monotone* type-2 fuzzy neural network was reported in (Li, Yi, Wang, & Zhang, 2013). A total of 567 input-output data pairs are obtained from (Fanger, 1970). Among them, 356 input-output data pairs (63%) are randomly selected as the training data set, as presented in Figure 5.1. The remaining 211 input-output data pairs (37%) are used as the validation data set, as presented in Figure 5.2. Besides that, the 356 input-output training data pairs and 211 input-output validation data pairs are injected with 30% and 50% noise levels by using Equation 5.1

$$Noise_{data} = actual_{data} + \left(\frac{random \times actual_{data} \times error}{100}\right) \tag{5.1}$$

where $actual_{data}$ is the clean (i.e., *monotone*) training data sample, $random$ is a normally distributed random number with mean 0 and variance 1, and the error setting is either 30% or 50%. The noisy (i.e., *non-monotone*) training and validation data sets for 30% noise are illustrated in Figure 5.3 and Figure 5.4, respectively. While, the noisy (i.e., *non-monotone*) training and validation data sets for 50% noise are illustrated in Figure 5.5 and Figure 5.6, respectively.

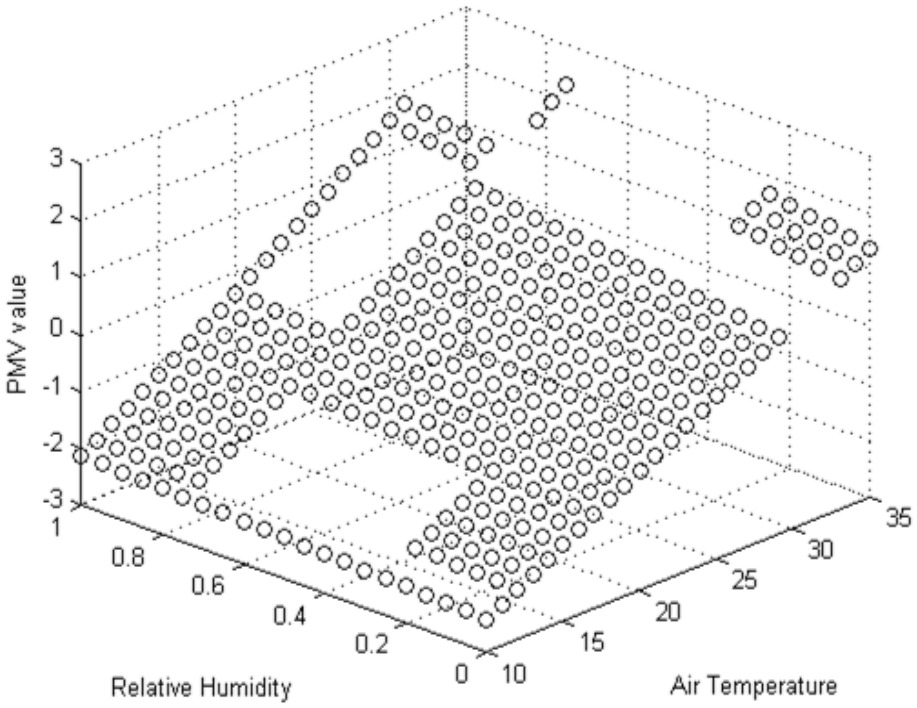

Figure 5.1: Training data of the thermal comfort index prediction without noise

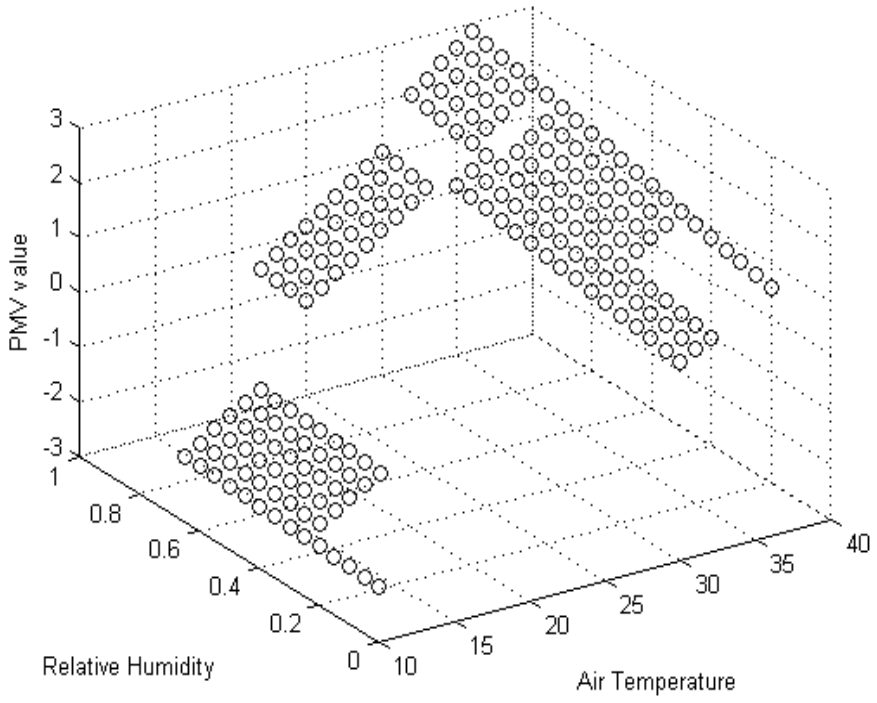

Figure 5.2: Validation data of the thermal comfort index prediction without noise

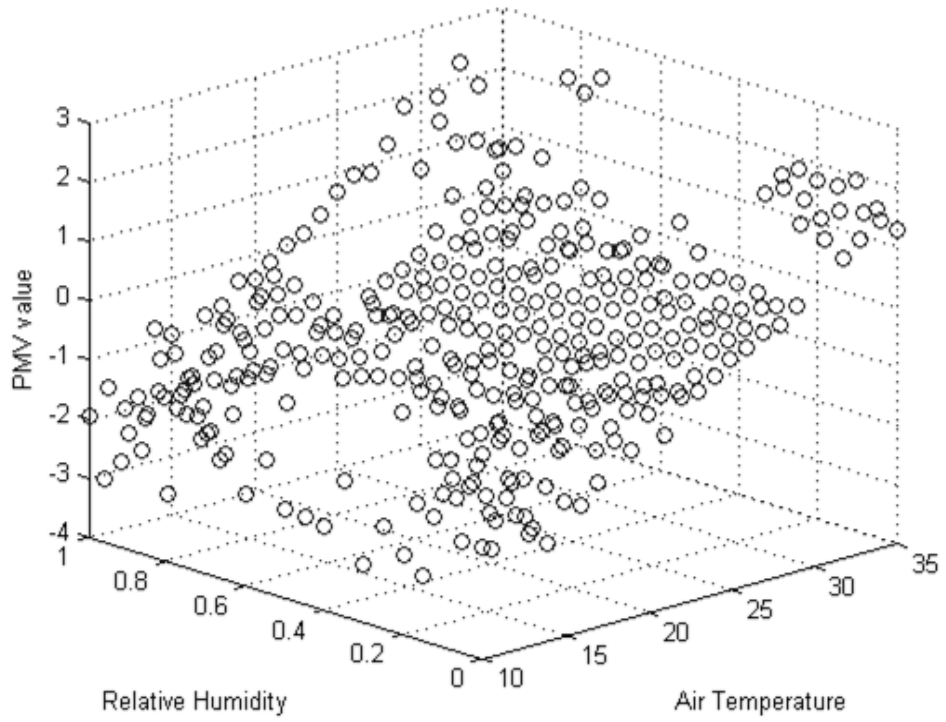

Figure 5.3: Training data of the thermal comfort index prediction for 30% noise

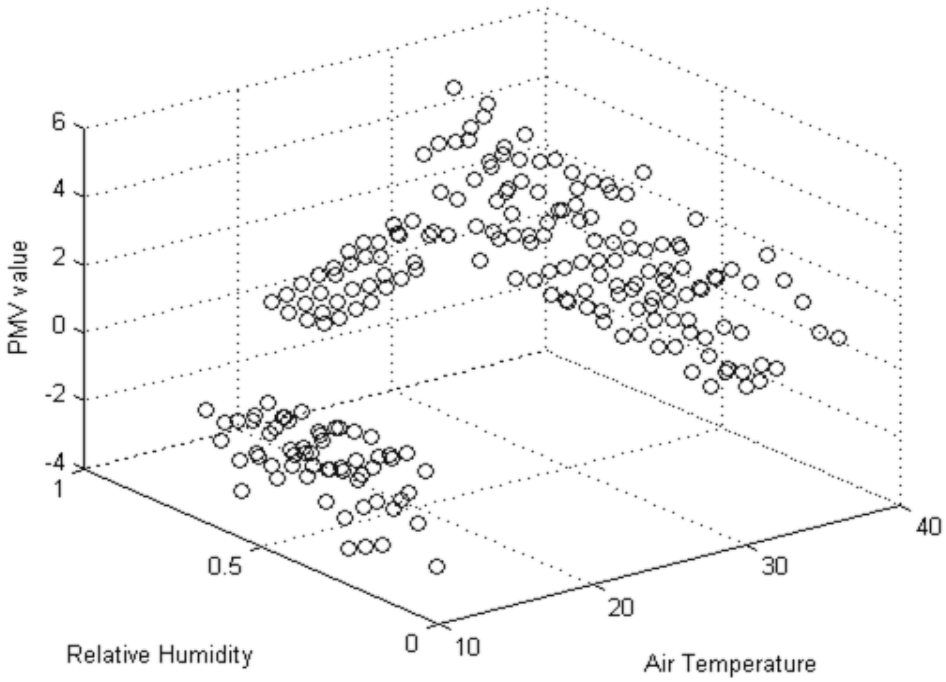

Figure 5.4: Validation data of the thermal comfort index prediction for 30% noise

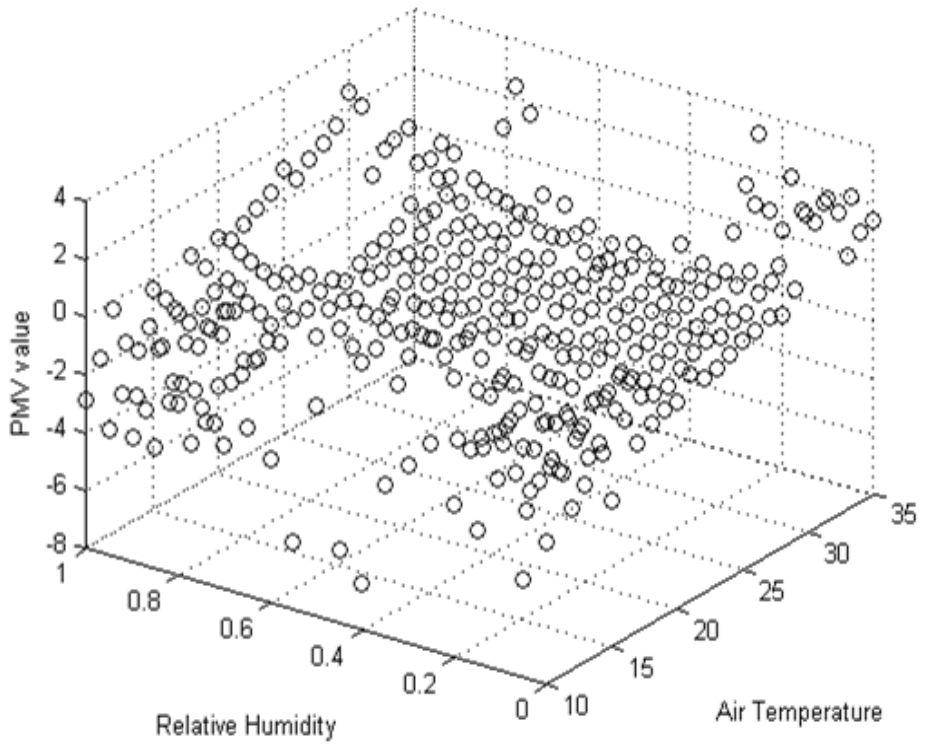

Figure 5.5: Training data of the thermal comfort index prediction for 50% noise

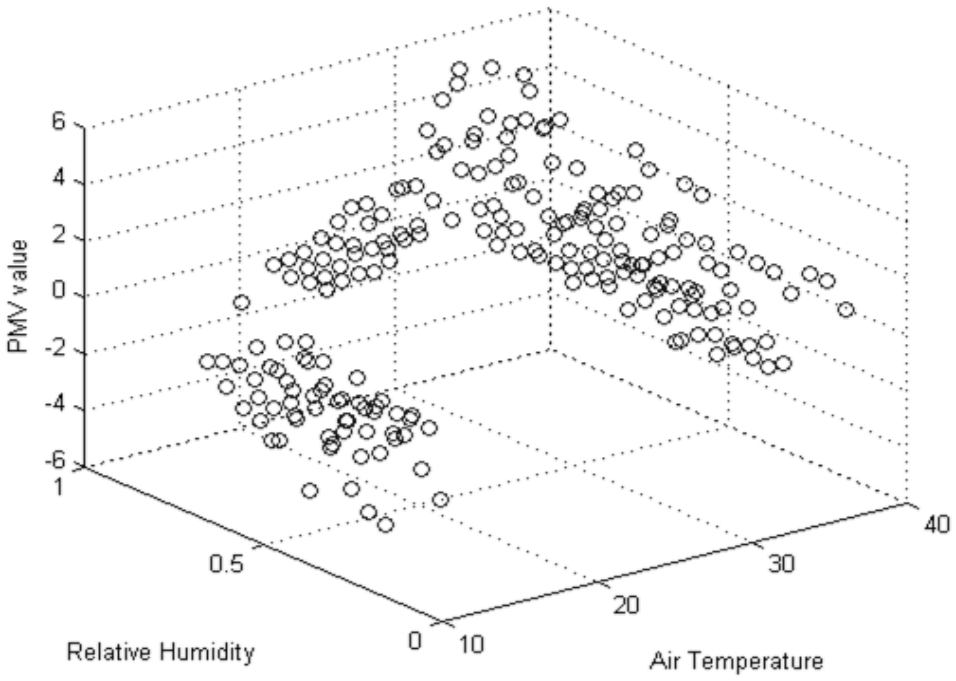

Figure 5.6: Validation data of the thermal comfort index prediction for 50% noise

The fuzzy partitions for $k_1 = k_2 = 3$ using trapezoidal and triangular fuzzy set are illustrated in Figure 5.7 and Figure 5.8, respectively. The fuzzy partitions for $k_1 = k_2 = 5$ using trapezoidal and triangular fuzzy set are illustrated in Figure 5.9 and Figure 5.10, respectively. The fuzzy partitions for $k_1 = k_2 = 7$ using trapezoidal and triangular fuzzy set are illustrated in Figure 5.11 and Figure 5.12, respectively.

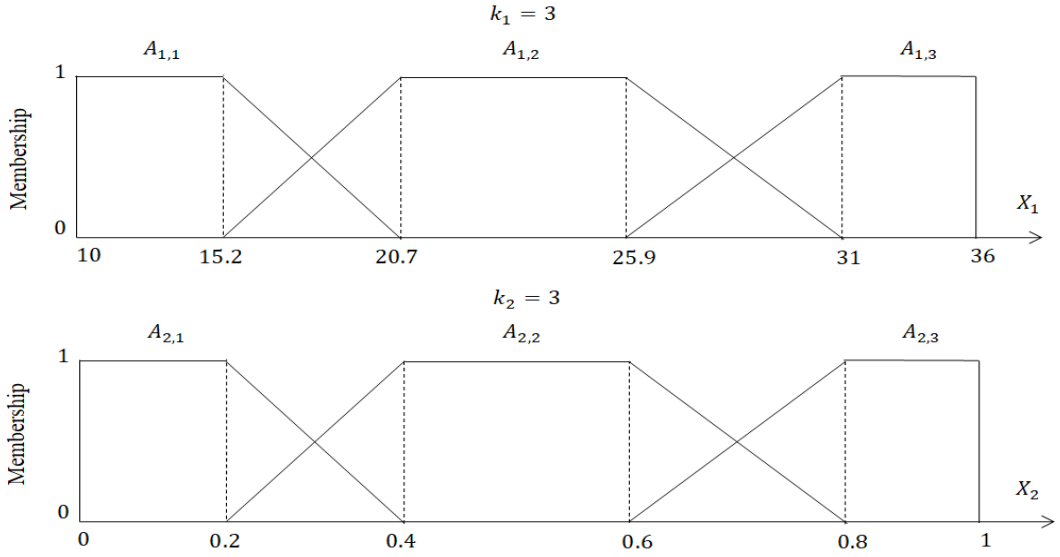

Figure 5.7: Fuzzy partition for $k_1 = k_2 = 3$ using trapezoidal fuzzy set

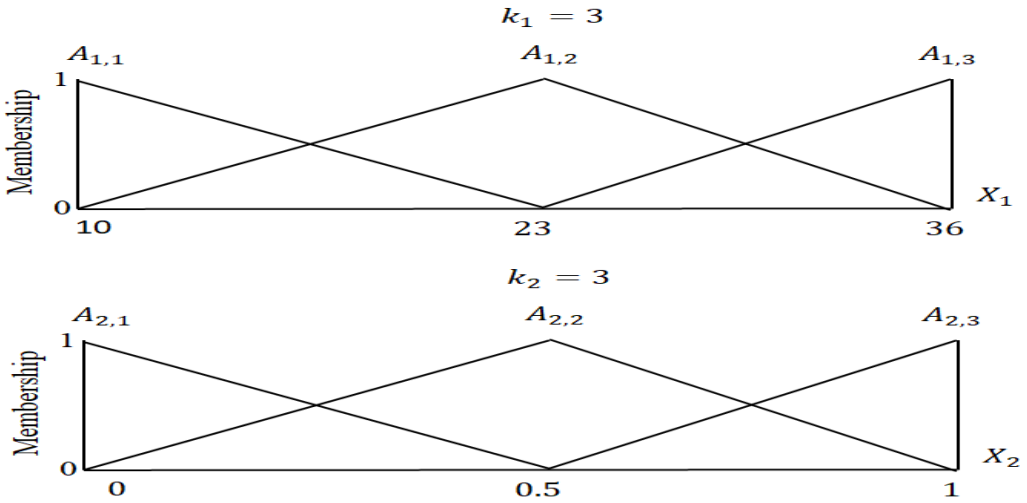

Figure 5.8: Fuzzy partition for $k_1 = k_2 = 3$ using triangular fuzzy set

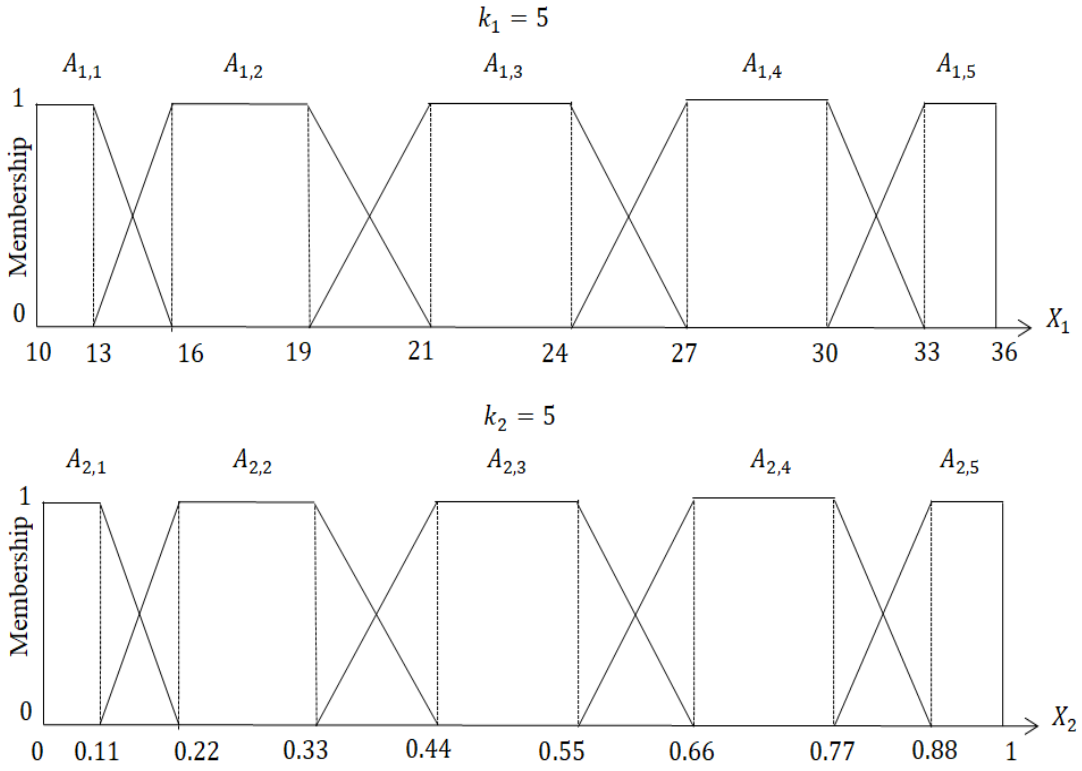

Figure 5.9: Fuzzy partition for $k_1 = k_2 = 5$ using trapezoidal fuzzy set

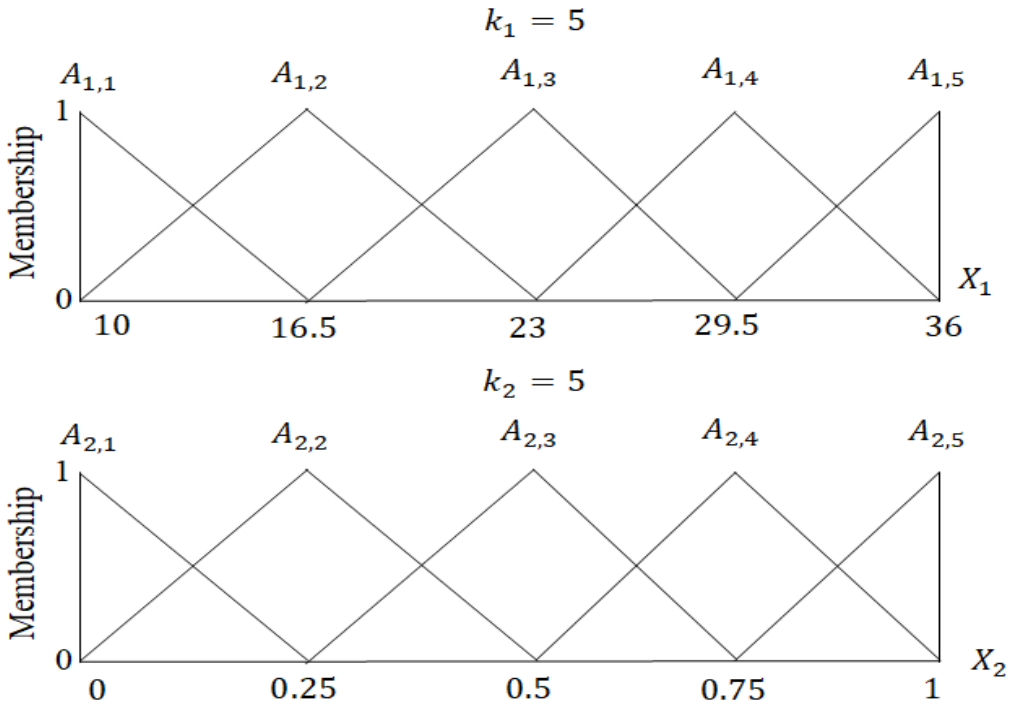

Figure 5.10: Fuzzy partition for $k_1 = k_2 = 5$ using triangular fuzzy set

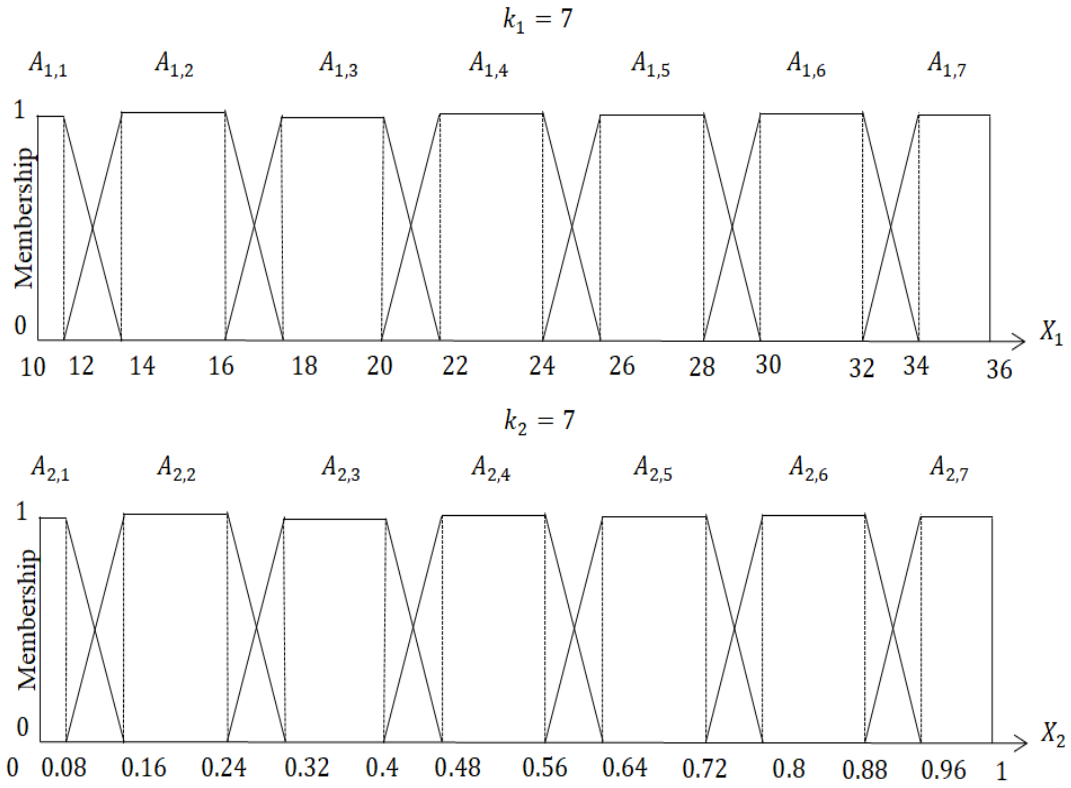

Figure 5.11: Fuzzy partition for $k_1 = k_2 = 7$ using trapezoidal fuzzy set

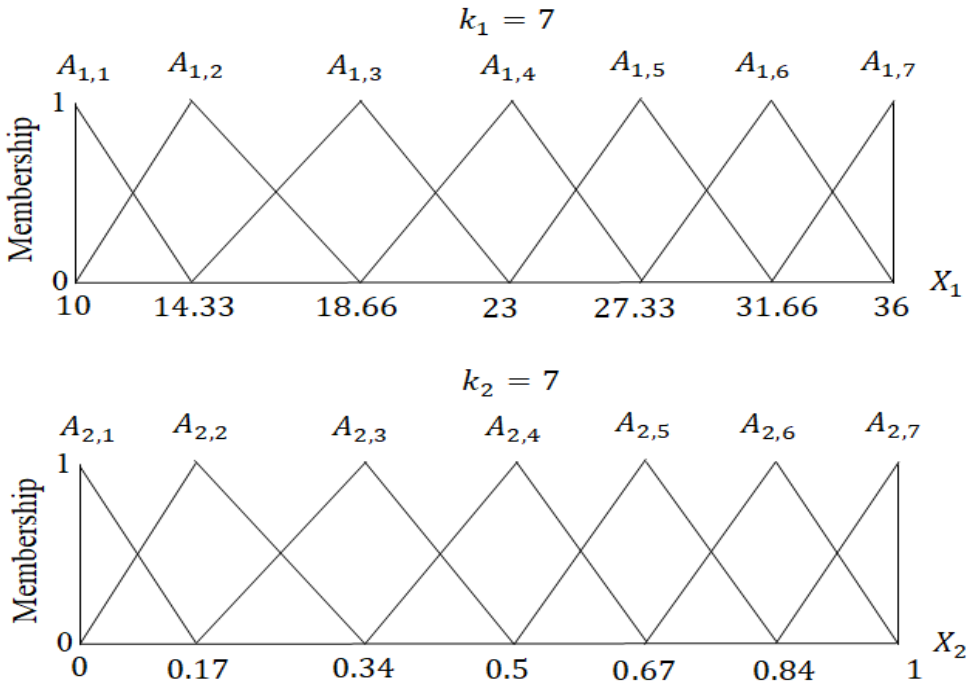

Figure 5.12: Fuzzy partition for $k_1 = k_2 = 7$ using triangular fuzzy set

5.2.2 Benchmark Example 2: Auto-Miles per Gallon (Auto-MPG)

A benchmark problem, i.e., Auto-MPG (Asuncion & Newman, 2007), is used for further evaluation. There are seven inputs attributes, i.e., cylinders, displacement, horsepower, weight, acceleration, model year and origin, and one output, i.e., auto-miles per gallon (Auto-MPG) values. Auto-MPG has a *monotone* relationship with two input attributes, namely horsepower and weight. When the horsepower and/or the weight attribute increase, Auto-MPG decreases. A total of 392 data samples are available. Among them, 228 input-output pairs (58%) are randomly selected as the training data, as presented in Figure 5.13. The remaining 164 input-output pairs (42%) are used as the validation data, as presented in Figure 5.14.

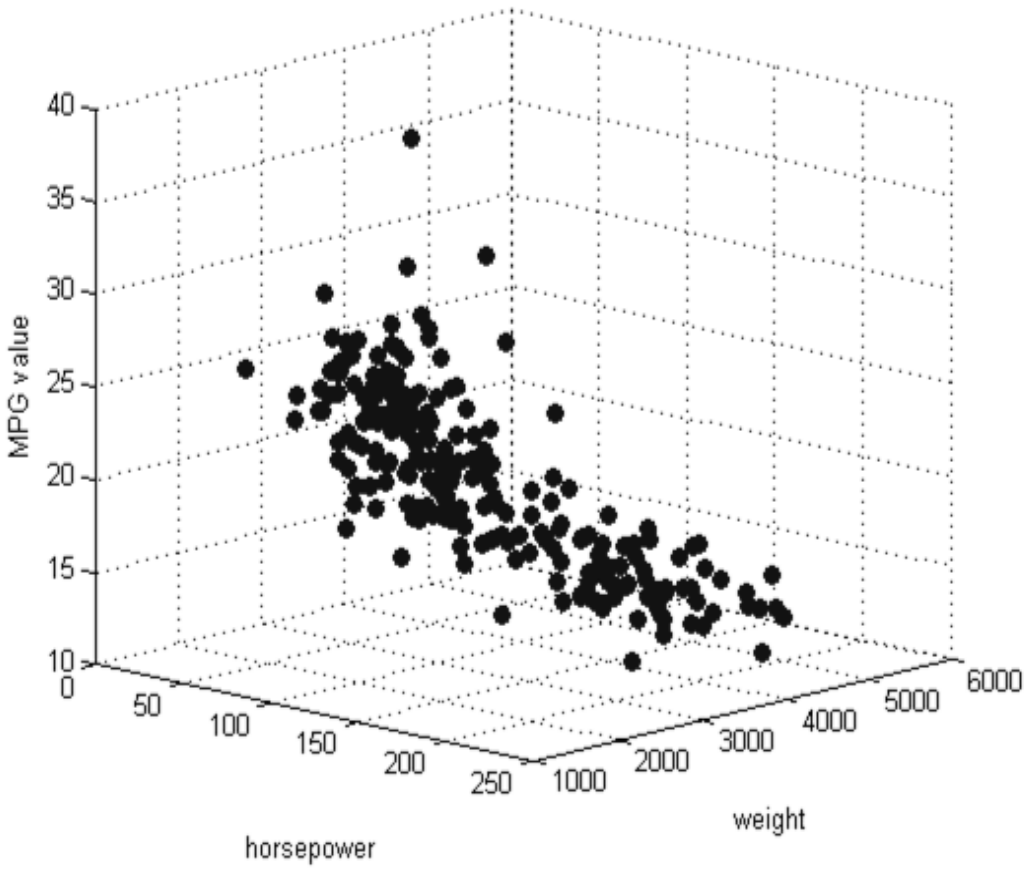

Figure 5.13: Training data of the Auto-MPG problem

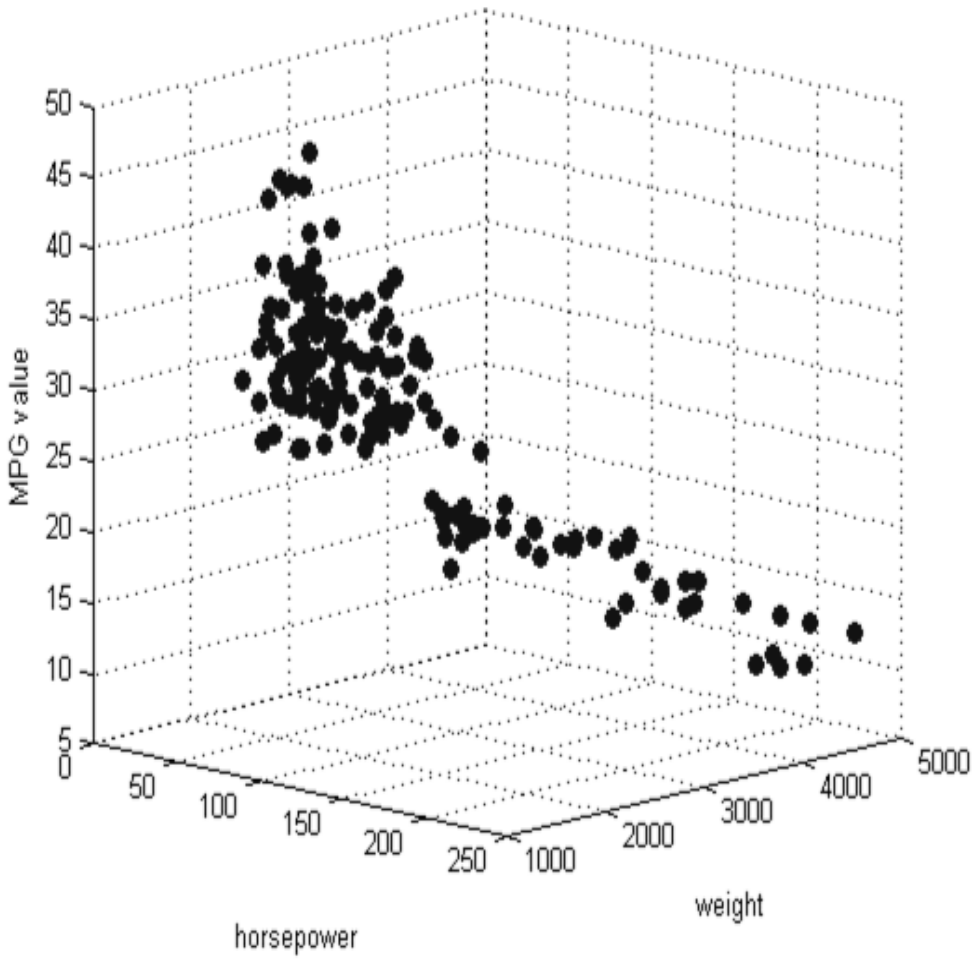

Figure 5.14: Validation data of the Auto-MPG problem

The fuzzy partitions for $k_1 = k_2 = 3$ using trapezoidal and triangular fuzzy set are illustrated in Figure 5.15 and Figure 5.16, respectively. The fuzzy partitions for $k_1 = k_2 = 5$ using trapezoidal and triangular fuzzy set are illustrated in Figure 5.17 and Figure 5.18, respectively. The fuzzy partitions for $k_1 = k_2 = 7$ using trapezoidal and triangular fuzzy set are illustrated in Figure 5.19 and Figure 5.20, respectively.

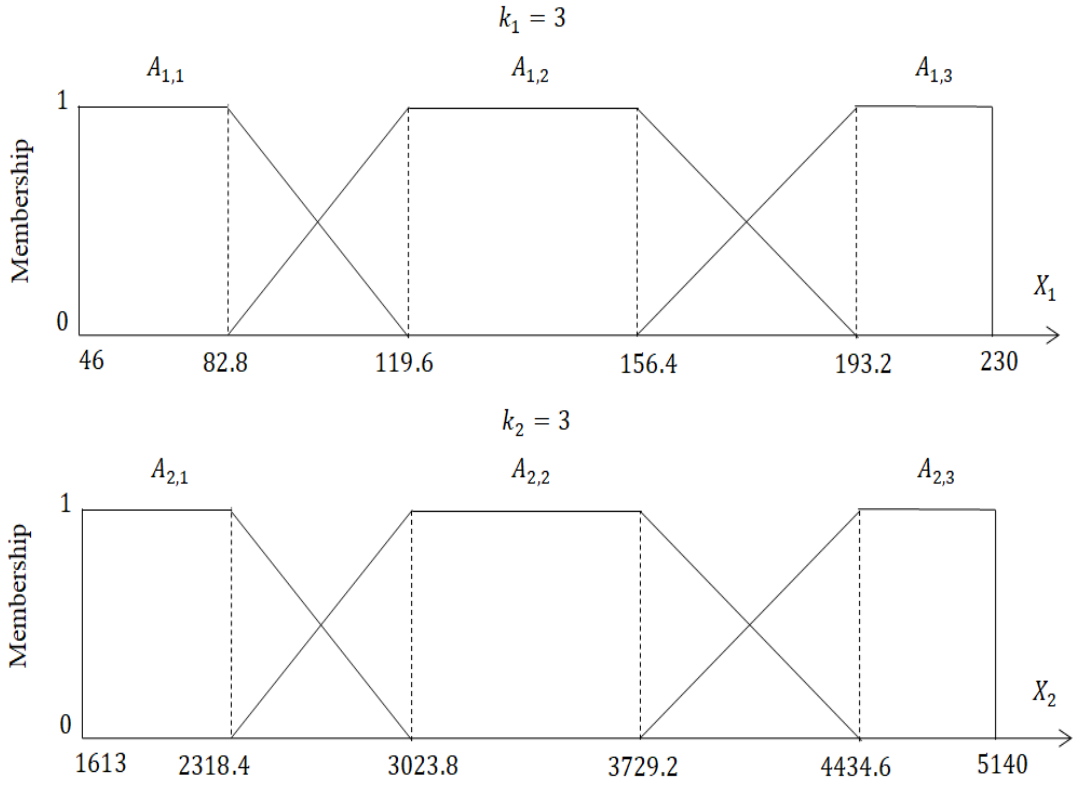

Figure 5.15: Fuzzy partition for $k_1 = k_2 = 3$ using trapezoidal fuzzy set

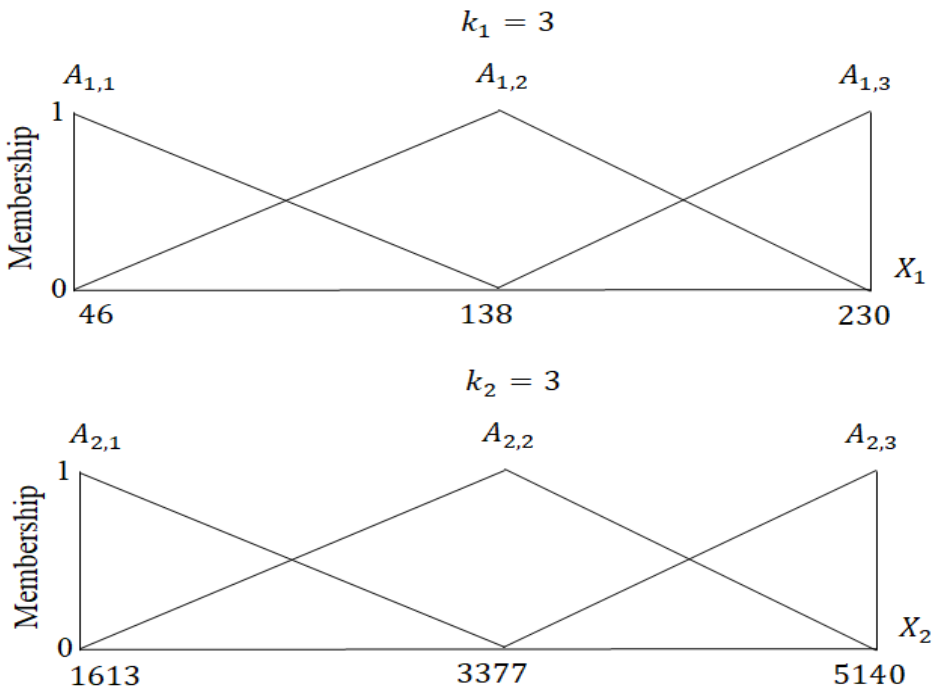

Figure 5.16: Fuzzy partition for $k_1 = k_2 = 3$ using triangular fuzzy set

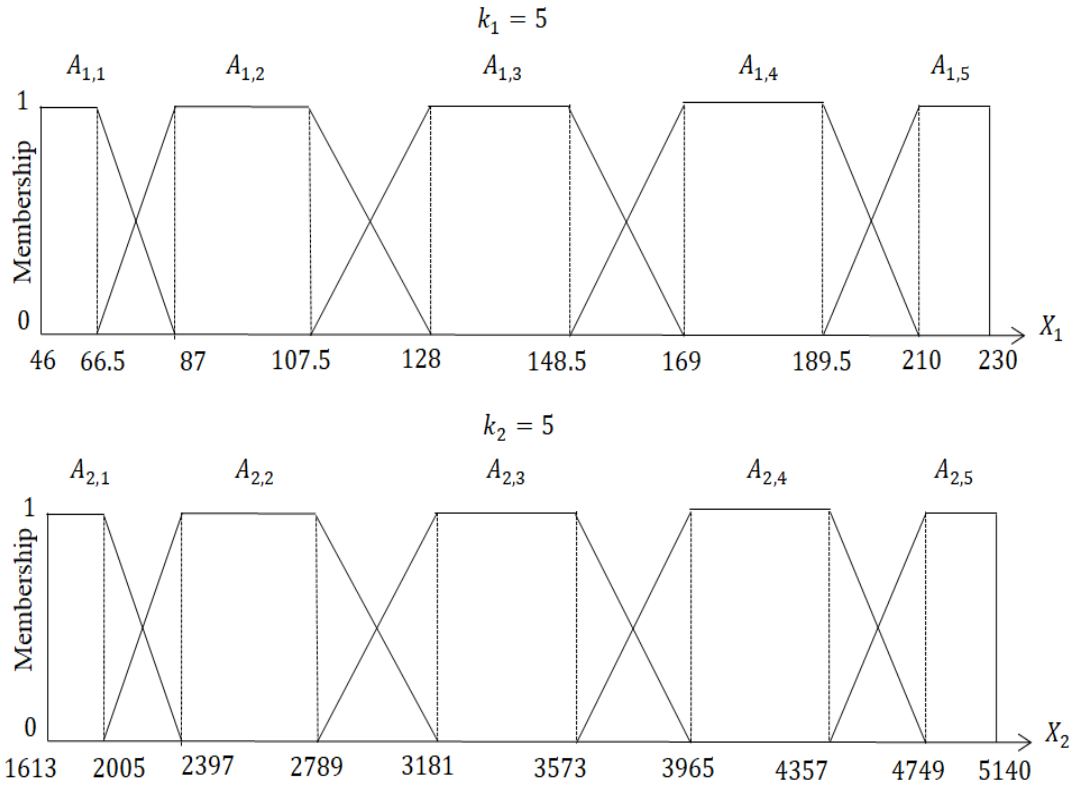

Figure 5.17: Fuzzy partition for $k_1 = k_2 = 5$ using trapezoidal fuzzy set

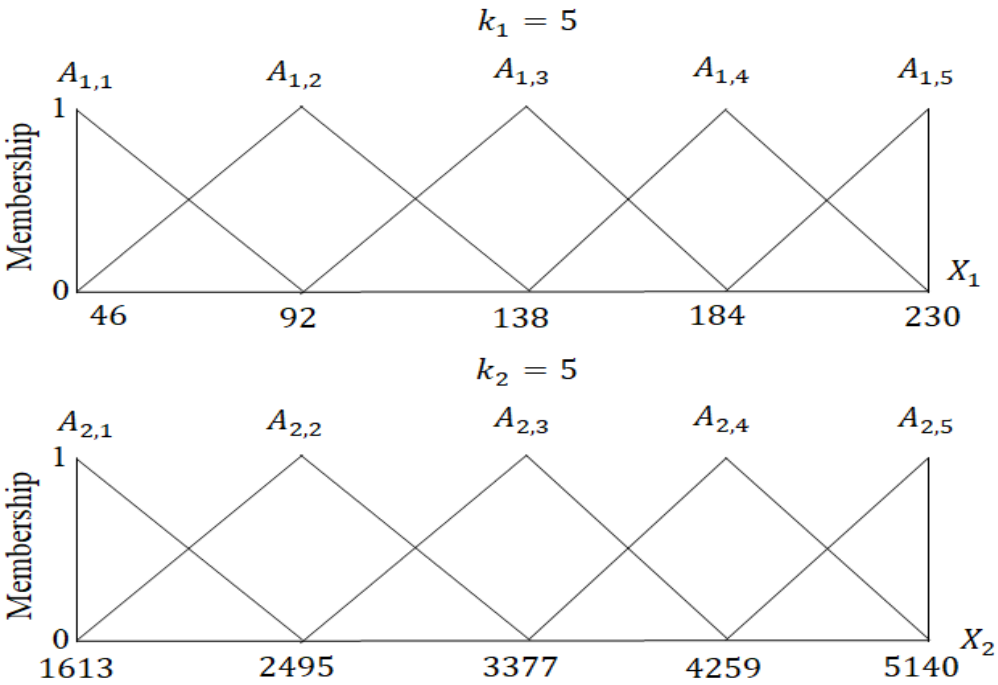

Figure 5.18: Fuzzy partition for $k_1 = k_2 = 5$ using triangular fuzzy set

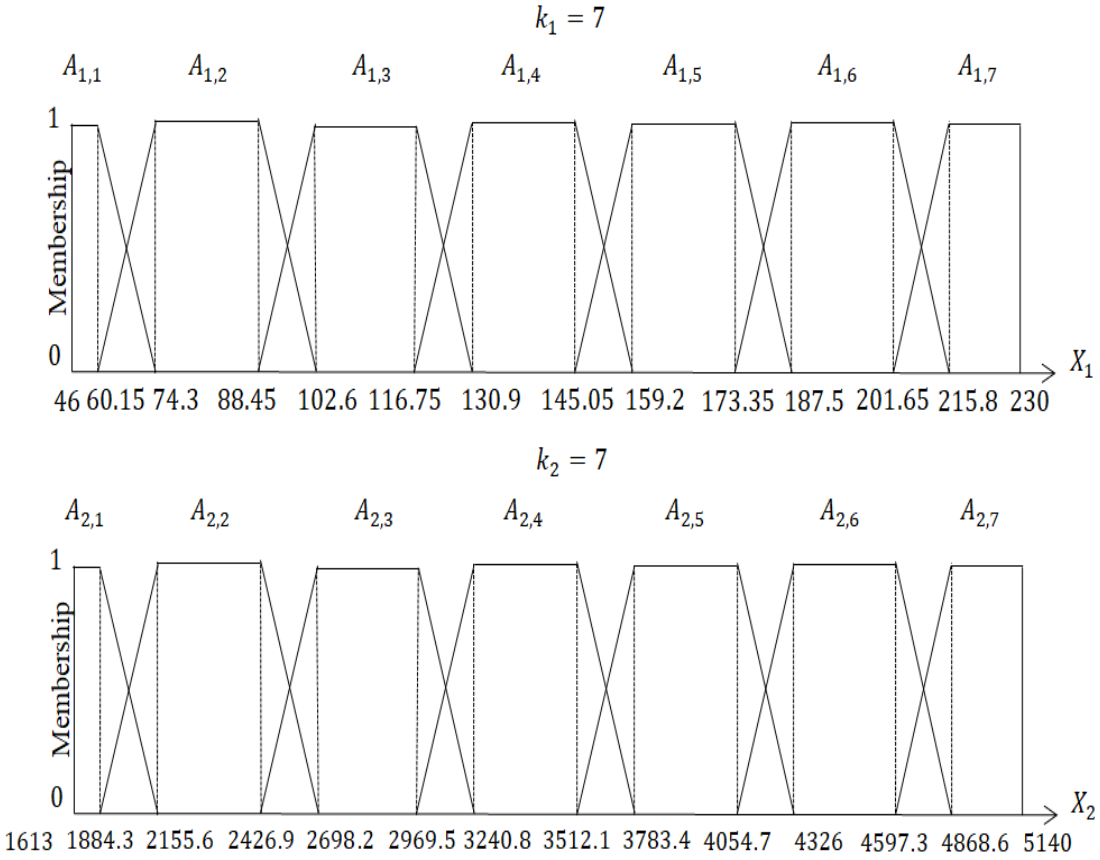

Figure 5.19: Fuzzy partition for $k_1 = k_2 = 7$ using trapezoidal fuzzy set

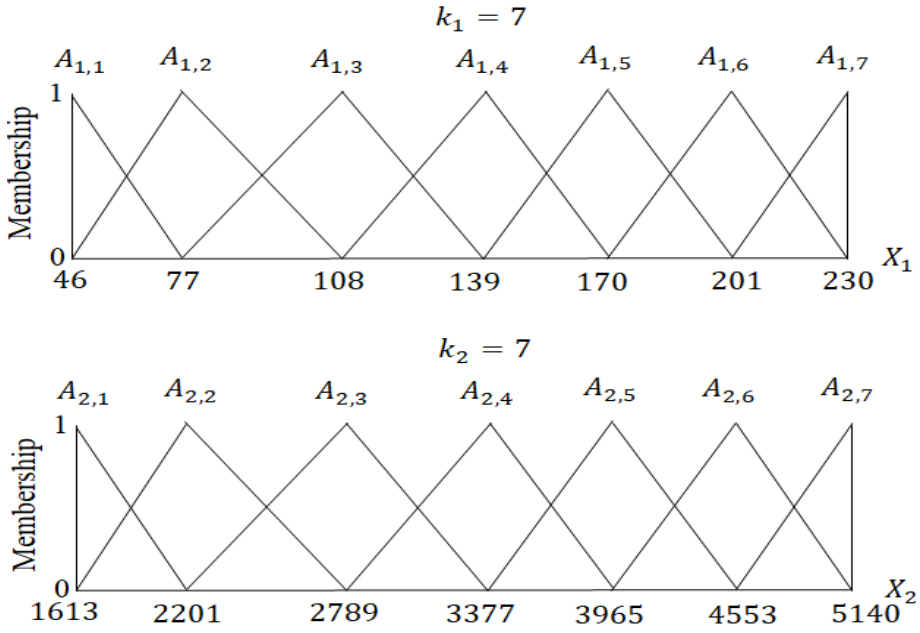

Figure 5.20: Fuzzy partition for $k_1 = k_2 = 7$ using triangular fuzzy set

5.2.3 Benchmark Example 3: Mercury-in-Fish

This problem is concerned with mercury contaminant in Atlantic cod (Gadu morhua) caught at the location of $68^{\circ}12'0"$N, $14^{\circ}48'0"$E, North Atlantic. There are 7 input attributes, namely sampling year, sampling site (i.e., latitude and longitude), age, body length, sex, sample ID, measurement ID, and the output is the mercury level in the fish muscle. The mercury concentration monotonically increases with respect to increasing body length and age of fish. A total of 467 input-output data pairs from ("International Council for the Exploration of the Sea (ICES)," n.d.) are used. Among them, 280 data pairs (60%) are randomly as the training data set, as presented in Figure 5.21. The remaining 187 data pairs (40%) are used as the validation data set, as presented in Figure 5.22.

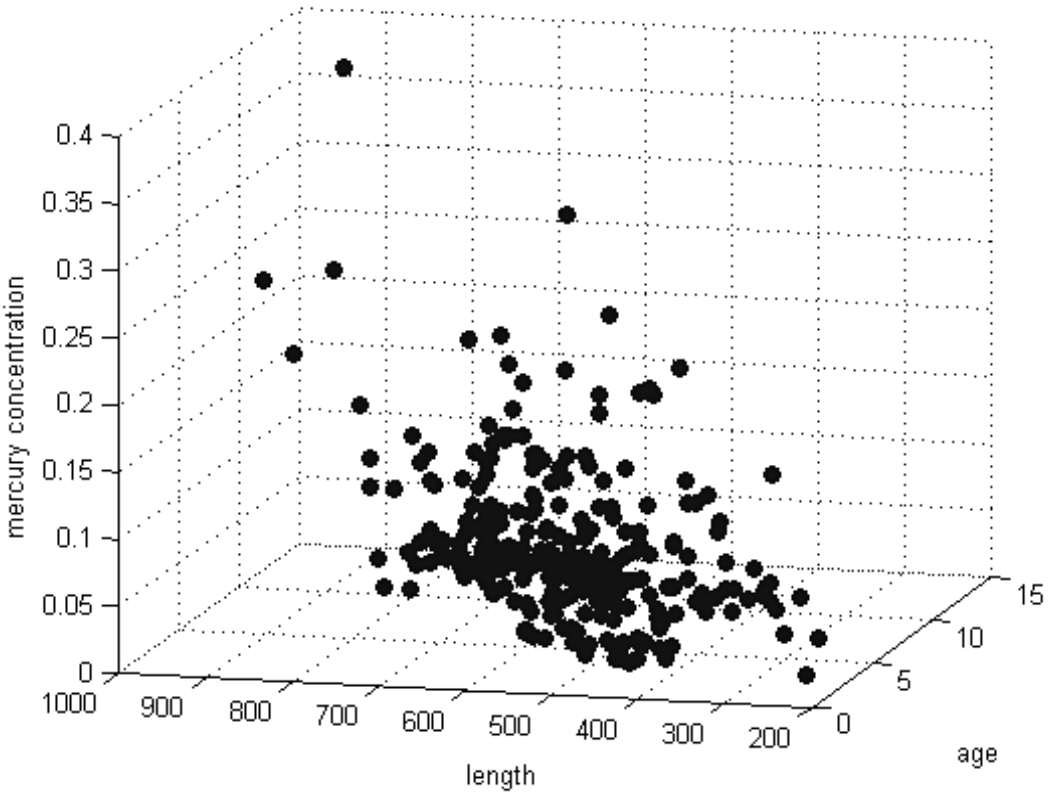

Figure 5.21: Training data of the mercury-in-fish problem

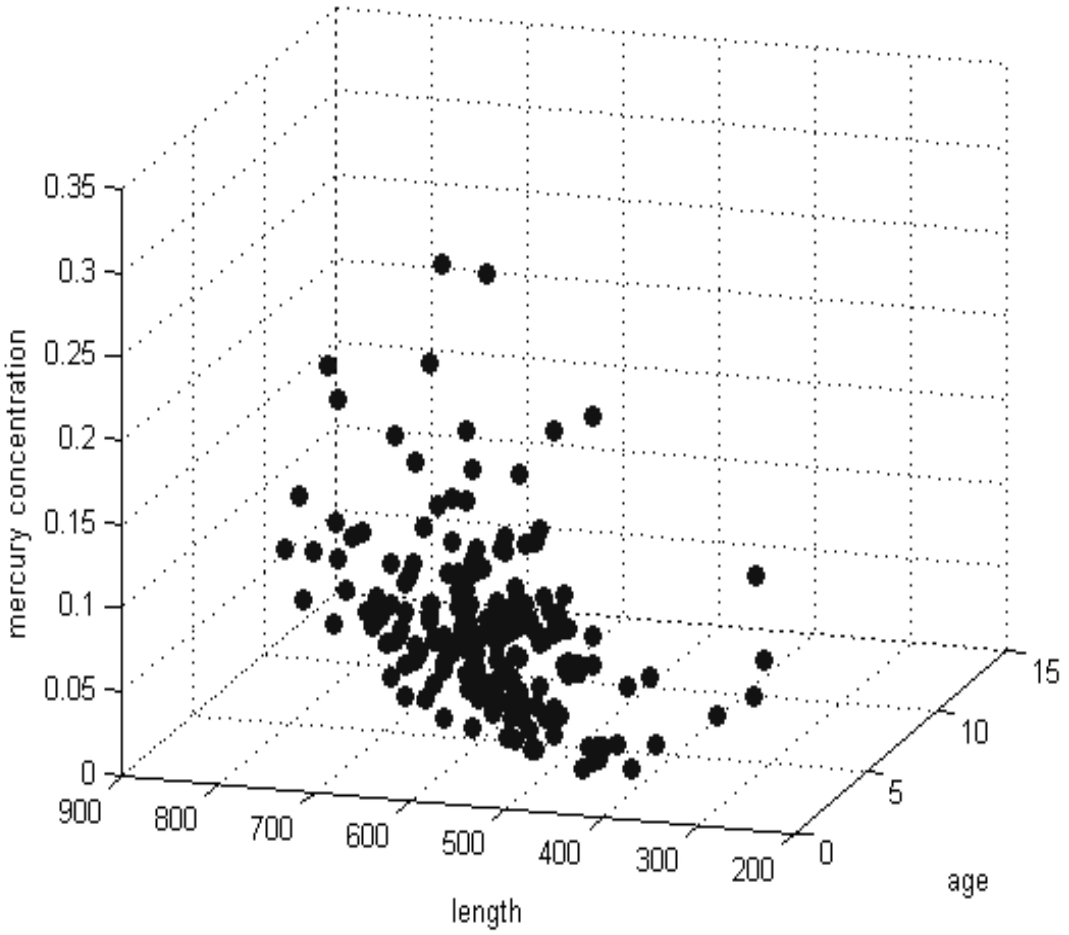

Figure 5.22: Validation data of the mercury-in-fish problem

The fuzzy partitions for $k_1 = k_2 = 3$ using trapezoidal and triangular fuzzy set are illustrated in Figure 5.23 and Figure 5.24, respectively. The fuzzy partitions for $k_1 = k_2 = 5$ using trapezoidal and triangular fuzzy set are illustrated in Figure 5.25 and Figure 5.26, respectively. The fuzzy partitions for $k_1 = k_2 = 7$ using trapezoidal and triangular fuzzy set are illustrated in Figure 5.27 and Figure 5.28, respectively.

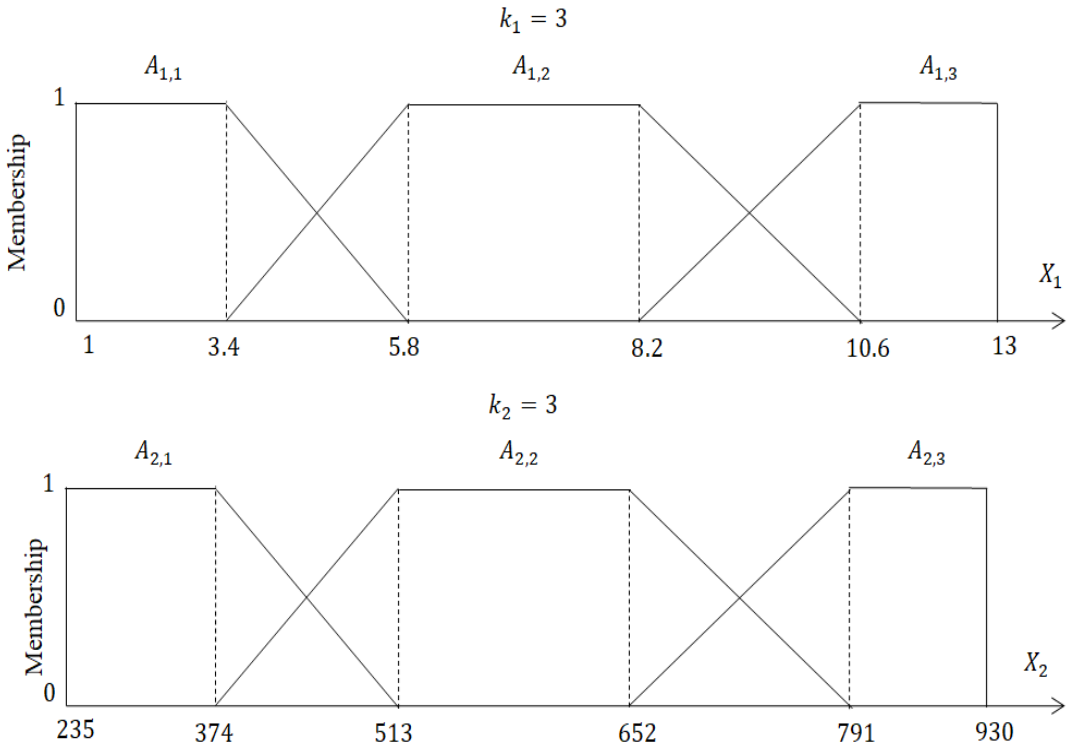

Figure 5.23: Fuzzy partition for $k_1 = k_2 = 3$ using trapezoidal fuzzy set

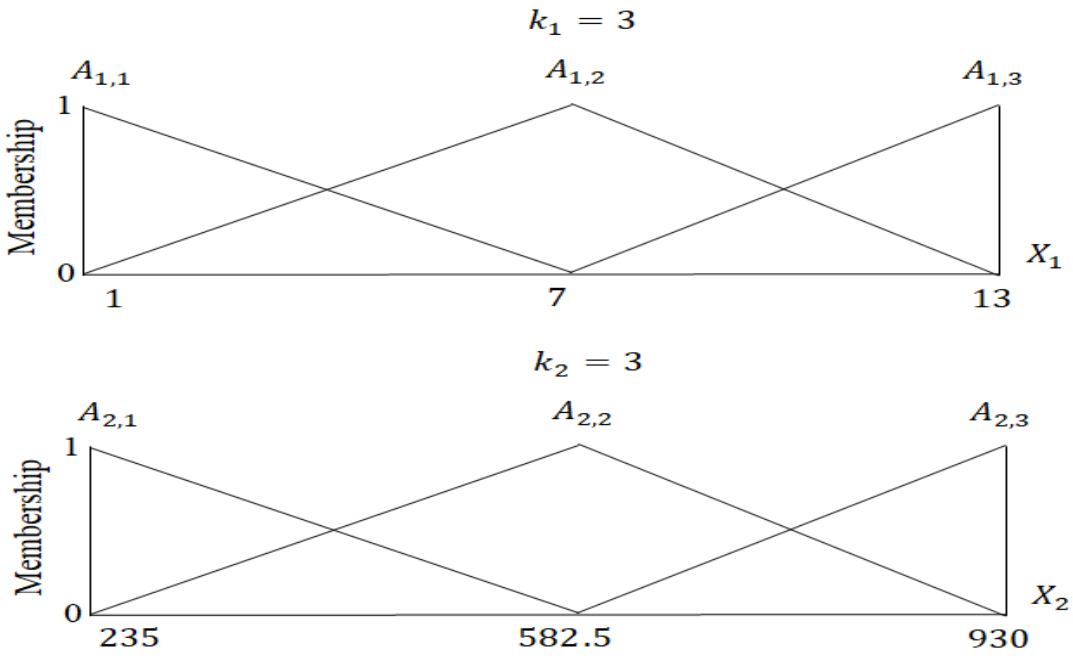

Figure 5.24: Fuzzy partition for $k_1 = k_2 = 3$ using triangular fuzzy set

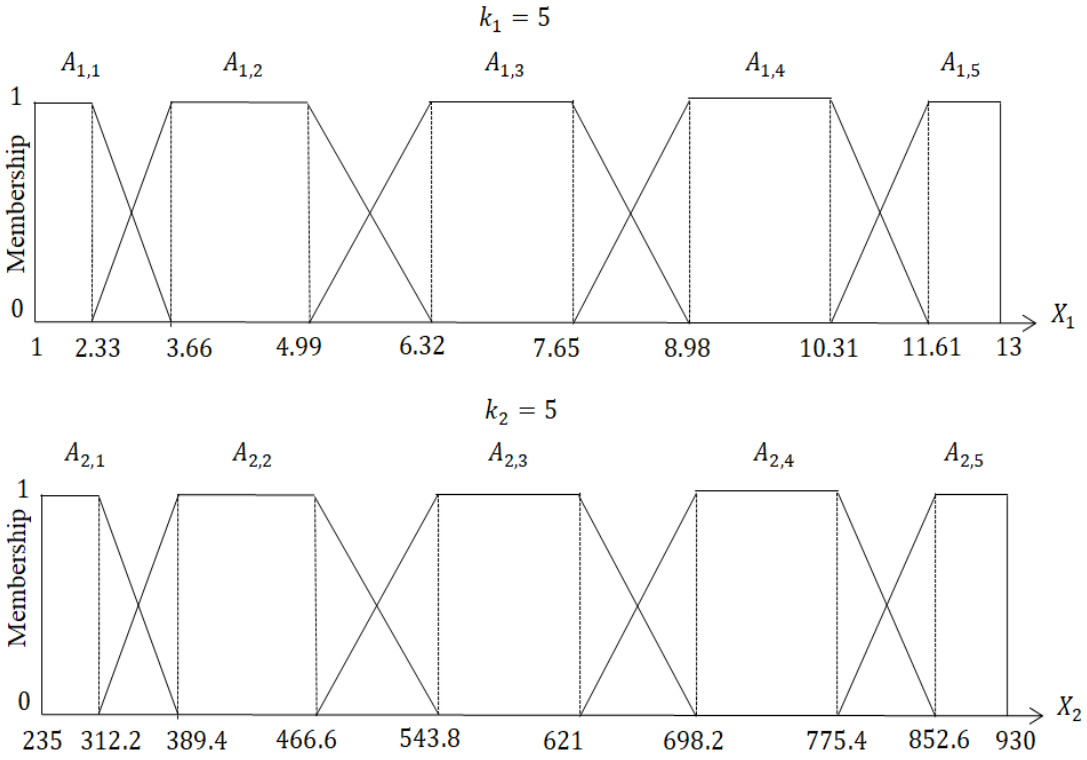

Figure 5.25: Fuzzy partition for $k_1 = k_2 = 5$ using trapezoidal fuzzy set

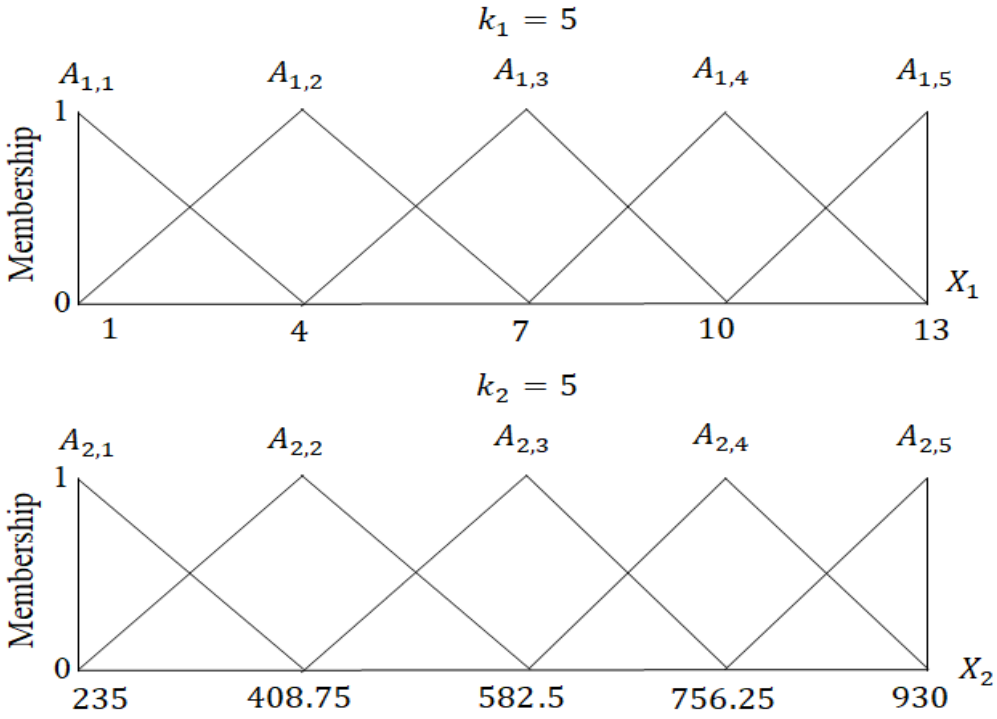

Figure 5.26: Fuzzy partition for $k_1 = k_2 = 5$ using triangular fuzzy set

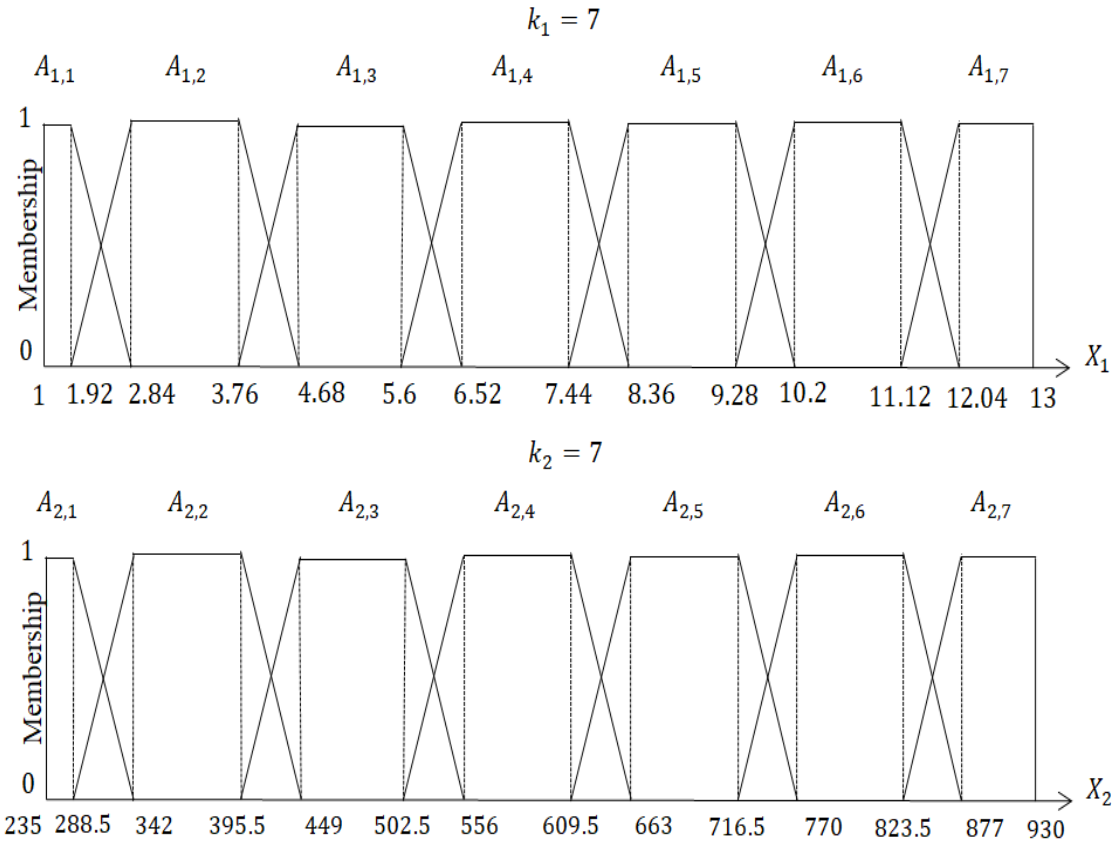

Figure 5.27: Fuzzy partition for $k_1 = k_2 = 7$ using trapezoidal fuzzy set

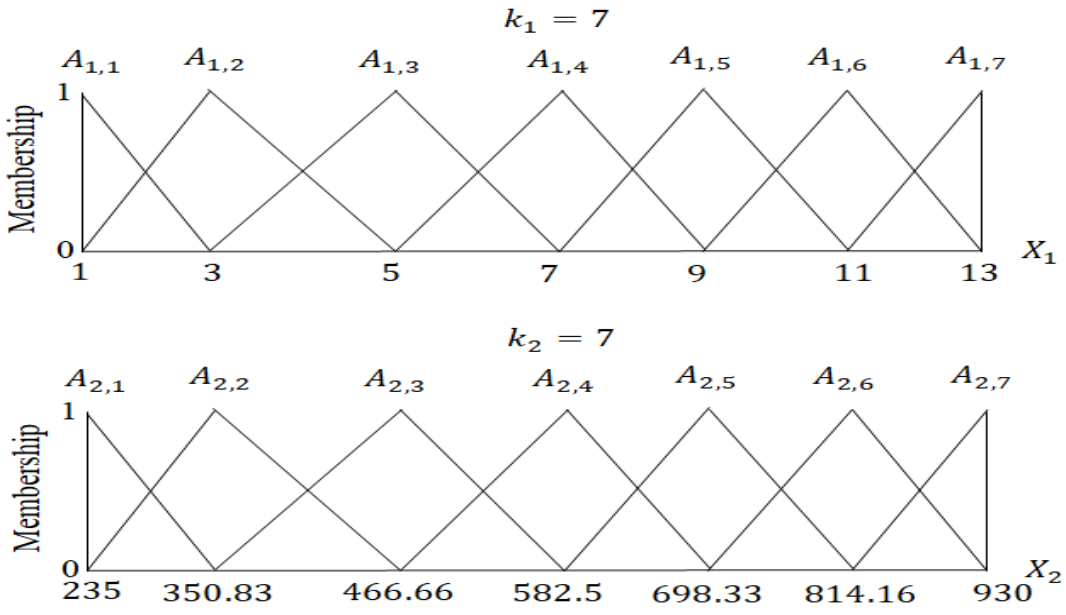

Figure 5.28: Fuzzy partition for $k_1 = k_2 = 7$ using triangular fuzzy set

5.3 Results and Discussions

5.3.1 Benchmark Example 1: Thermal Comfort Heat Index Prediction

The DOM scores of the resulting FIS models are measured using Equation 3.5 (see Chapter 3 Section 3.4), in which $N_1 = N_2 = 26$ is used. The results for thermal comfort index prediction with WM and Ishibuchi methods using trapezoidal fuzzy set are depicted in Table 5.1 and Table 5.2, respectively. In Table 5.1 (a), for $k_1 = k_2 = 3$ (see Figure 5.7), *complete* and *non-monotone* fuzzy If-Then rules are obtained using WM method. Therefore, the WFRI (i.e., Stage 4 in Figure 4.17) is omitted as the is no missing fuzzy rule. On the other hand, as depicted in Table 5.1 (b), for $k_1 = k_2 = 5$ (see Figure 5.9) and Table 5.1 (c), for $k_1 = k_2 = 7$ (see Figure 5.11), columns "*Without relabeling*" and "*With relabeling*" represent the SSE and DOM scores for the FIS models, based on the fuzzy rules obtained after WFRI and MFRR (i.e., Stage 4 and 5 in Figure 4.17), respectively. Column "*Strategies*" represents the four weight assignment strategies with Equation 4.5 in weight assignment (i.e., Stage 3 in Figure 4.17), based on weight normalization. As an example, in Table 5.1 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.8571 and 1.0000, the SSE values with the training data set are 10.9377 and 9.1769, the SSE values with the validation data set are 138.8669 and 15.1793 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing MFRR.

Table 5.1: Experiment result for WM method using trapezoidal fuzzy set (a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.8929	29.9790	61.4486
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	30.1029	52.6213
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	27.3098	42.0002
4	w_{j_1,j_2}^{sum}	1.0000	27.2619	41.6924
5	$w_{j_1,j_2}^{\max}$	1.0000	26.8402	38.8893

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.9080	9.3546	30.7457	1.0000	9.0287	13.8105
2	$w_{j_1,j_2}^{\text{data}}$	0.8571	10.9377	138.8669	1.0000	9.1769	15.1793
3	w_{j_1,j_2}^{sum}	0.8613	10.8155	129.5768	1.0000	9.1862	15.4330
4	$w_{j_1,j_2}^{\max}$	0.9080	9.3546	30.7457	1.0000	9.0287	13.8105

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8434	5.6987	43.4119	1.0000	5.3578	11.4854
2	$w_{j_1,j_2}^{\text{data}}$	0.8063	7.0094	142.8811	1.0000	5.2288	12.5533
3	w_{j_1,j_2}^{sum}	0.8063	6.9862	140.0785	1.0000	5.1835	10.8268
4	$w_{j_1,j_2}^{\max}$	0.8310	6.0467	68.9995	1.0000	5.2304	10.6106

(c)

Similar results are obtained using the Ishibuchi method for trapezoidal fuzzy set. As an example, in Table 5.2 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.8791 and 1.0000, the SSE values with the training data set are 9.8525 and 10.1573, the SSE values with the validation data set are 71.9501 and 21.4568 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR. Again, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing MFRR.

Table 5.2: Experiment result for Ishibuchi method using trapezoidal fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.8929	38.3973	70.0695
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	38.7553	65.3484
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	38.7890	65.3707
4	w_{j_1,j_2}^{sum}	1.0000	36.6320	57.5382
5	w_{j_1,j_2}^{max}	1.0000	36.4123	55.4779

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.9080	9.8525	25.0090	1.0000	9.8935	21.8471
2	$w_{j_1,j_2}^{\text{data}}$	0.8791	9.8525	71.9501	1.0000	10.1573	21.4568
3	w_{j_1,j_2}^{sum}	0.8832	9.8525	67.6524	1.0000	9.9414	17.6576
4	w_{j_1,j_2}^{max}	0.9025	9.8525	28.1124	1.0000	9.6679	17.1629

(b)

Table 5.2 continued

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8997	5.2940	21.1717	1.0000	5.3360	13.4033
2	$w_{j_1,j_2}^{\text{data}}$	0.8846	5.2940	74.0983	1.0000	5.2285	11.4332
3	w_{j_1,j_2}^{sum}	0.8846	5.2940	76.1265	1.0000	5.2176	10.6164
4	w_{j_1,j_2}^{max}	0.8887	5.2940	47.1595	1.0000	5.2541	13.2675

(c)

The results for the WM and Ishibuchi methods using triangular fuzzy set are depicted in Table 5.3 and Table 5.4, respectively. As an example, in Table 5.3 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.8077 and 1.0000, the SSE values with the training data set are 5.9010 and 4.8308, the SSE values with the validation data set are 95.4047 and 16.6386 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing MFRR.

Table 5.3: Experiment result for WM method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.6607	17.6087	34.5017
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	17.2273	34.8156
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	18.9862	37.6318
4	w_{j_1,j_2}^{sum}	1.0000	19.6494	40.3102
5	w_{j_1,j_2}^{max}	1.0000	18.5832	40.4202

(a)

Table 5.3 continued

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.9052	5.1294	26.9059	1.0000	4.7538	14.5480
2	$w_{j_1,j_2}^{\text{data}}$	0.8077	5.9010	95.4047	1.0000	4.8308	16.6386
3	w_{j_1,j_2}^{sum}	0.8077	5.8855	94.2250	1.0000	4.8045	17.2898
4	w_{j_1,j_2}^{max}	0.8681	5.2360	35.9842	1.0000	4.7880	14.9895

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7995	1.5438	27.1411	1.0000	0.7773	5.6782
2	$w_{j_1,j_2}^{\text{data}}$	0.6690	6.5720	148.0971	1.0000	0.9275	2.7826
3	w_{j_1,j_2}^{sum}	0.6676	6.3635	149.5647	1.0000	0.9587	3.3607
4	w_{j_1,j_2}^{max}	0.7486	3.2139	78.1586	1.0000	0.8755	7.5532

(c)

Similar results are obtained using the Ishibuchi method for triangular fuzzy set. As an example, in Table 5.4 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.7418 and 1.0000, the SSE values with the training data set are 9.0861 and 9.1212, the SSE values with the validation data set are 73.1856 and 30.6314 after WFRI and MFRR, respectively. Again, the SSE values for both training and validation data set are better (i.e., lower) with the addition of weight assignment to MFRR, as compared with those after *ad hoc* methods.

Table 5.4: Experiment result for Ishibuchi method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.9821	67.0466	138.5177
2	$w_{One-Zero}^{j_1,j_2}$	1.0000	67.0187	137.9639
3	$w_{data}^{j_1,j_2}$	1.0000	67.2722	138.6570
4	$w_{sum}^{j_1,j_2}$	1.0000	67.0203	137.9591
5	$w_{max}^{j_1,j_2}$	1.0000	66.6931	137.0501

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{One-Zero}^{j_1,j_2}$	0.8242	9.0861	30.9672	1.0000	8.7286	30.4533
2	$w_{data}^{j_1,j_2}$	0.7418	9.0861	73.1856	1.0000	9.1212	30.6314
3	$w_{sum}^{j_1,j_2}$	0.7418	9.0861	72.4165	1.0000	8.5634	26.4114
4	$w_{max}^{j_1,j_2}$	0.7761	9.0861	39.9889	1.0000	8.3047	25.5287

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{One-Zero}^{j_1,j_2}$	0.7706	2.5918	21.8914	1.0000	2.8271	13.5027
2	$w_{data}^{j_1,j_2}$	0.7473	2.5918	64.8075	1.0000	2.3879	10.2728
3	$w_{sum}^{j_1,j_2}$	0.7473	2.5918	64.1859	1.0000	2.1771	9.3984
4	$w_{max}^{j_1,j_2}$	0.7610	2.5918	44.9824	1.0000	2.3397	12.1379

(c)

The results for thermal comfort index prediction for 30% noise with WM and Ishibuchi methods using trapezoidal fuzzy set are depicted in Table 5.5 and Table 5.6, respectively. As an example, in Table 5.5 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.8049 and 1.0000, the SSE values with the training data set are 47.5880 and 49.8642, the SSE values with the validation data set are 223.3738 and 65.7381 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing MFRR.

Table 5.5: Experiment result for WM method using trapezoidal fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.8846	66.7310	137.4289
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	66.8464	124.1423
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	62.7304	109.9477
4	w_{j_1,j_2}^{sum}	1.0000	62.8444	110.6305
5	w_{j_1,j_2}^{max}	1.0000	62.0984	106.1629

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8558	47.3304	89.7535	1.0000	49.5149	63.1919
2	$w_{j_1,j_2}^{\text{data}}$	0.8049	47.5880	223.3738	1.0000	49.8642	65.7381
3	w_{j_1,j_2}^{sum}	0.8091	47.5323	211.9509	1.0000	49.8327	66.0533
4	w_{j_1,j_2}^{max}	0.8558	47.3304	89.7535	1.0000	49.5149	63.1919

(b)

Table 5.5 continued

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7596	39.0495	121.2256	1.0000	43.2958	60.5053
2	$w_{j_1,j_2}^{\text{data}}$	0.7225	40.6272	246.3257	1.0000	42.9992	62.4498
3	w_{j_1,j_2}^{sum}	0.7225	40.5898	243.0191	1.0000	42.9364	64.1229
4	w_{j_1,j_2}^{max}	0.7459	39.4540	154.6627	1.0000	42.9481	61.3318

(c)

Similar results are obtained using the Ishibuchi method for trapezoidal fuzzy set. As an example, in Table 5.6 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.8104 and 1.0000, the SSE values with the training data set are 47.7963 and 50.6245, the SSE values with the validation data set are 151.7245 and 88.3533 after WFRI and MFRR, respectively. Again, the SSE values for both training and validation data set are better (i.e., lower) with the addition of weight assignment to MFRR, as compared with those after *ad hoc* methods.

Table 5.6: Experiment result for Ishibuchi method using trapezoidal fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.8846	74.0996	144.6936
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	74.5055	139.5708
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	74.5037	139.5630
4	w_{j_1,j_2}^{sum}	1.0000	72.1989	131.4109
5	w_{j_1,j_2}^{max}	1.0000	71.8218	128.8089

(a)

Table 5.6 continued

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8393	47.7963	102.9497	1.0000	50.2421	89.0081
2	$w_{j_1,j_2}^{\text{data}}$	0.8104	47.7963	151.7245	1.0000	50.6245	88.3533
3	w_{j_1,j_2}^{sum}	0.8146	47.7963	146.7392	1.0000	50.1037	72.2929
4	w_{j_1,j_2}^{max}	0.8338	47.7963	105.2490	1.0000	49.7218	71.2200

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8255	38.7293	89.4617	1.0000	43.2602	67.3230
2	$w_{j_1,j_2}^{\text{data}}$	0.8063	38.7293	161.9113	1.0000	42.9372	58.2089
3	w_{j_1,j_2}^{sum}	0.8063	38.7293	164.6860	1.0000	42.9343	58.1760
4	w_{j_1,j_2}^{max}	0.8104	38.7293	126.5036	1.0000	42.9700	66.9248

(c)

The results for thermal comfort index prediction for 30% noise with WM and Ishibuchi methods using triangular fuzzy set are depicted in Table 5.7 and Table 5.8, respectively. As an example, in Table 5.7 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.6868 and 1.0000, the SSE values with the training data set are 43.5152 and 45.6128, the SSE values with the validation data set are 178.8184 and 73.8112 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing MFRR.

Table 5.7: Experiment result for WM method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.6332	57.0612	108.0212
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	56.6493	106.3600
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	62.2303	126.7408
4	w_{j_1,j_2}^{sum}	1.0000	62.9129	129.8240
5	$w_{j_1,j_2}^{\max}$	1.0000	60.1509	123.9425

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7843	43.2035	91.5111	1.0000	44.9760	74.0416
2	$w_{j_1,j_2}^{\text{data}}$	0.6868	43.5152	178.8184	1.0000	45.6128	73.8112
3	w_{j_1,j_2}^{sum}	0.6868	43.5063	177.3082	1.0000	45.4986	73.6766
4	$w_{j_1,j_2}^{\max}$	0.7418	43.2213	103.3218	1.0000	44.9706	72.1630

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.6511	38.0774	105.8859	1.0000	40.4069	69.9826
2	$w_{j_1,j_2}^{\text{data}}$	0.5426	43.0887	239.7809	1.0000	40.5825	62.1793
3	w_{j_1,j_2}^{sum}	0.5330	42.8675	241.3096	1.0000	40.4653	61.9567
4	$w_{j_1,j_2}^{\max}$	0.6071	39.5636	161.6221	1.0000	40.4794	75.6101

(c)

Similar results are obtained using the Ishibuchi method for triangular fuzzy set. As an example, in Table 5.8 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.6374 and 1.0000, the SSE values with the training data set are 47.1516 and 49.6366, the SSE values with the validation data set are 153.2022 and 98.5076 after WFRI and MFRR, respectively. Again, the SSE values for both training and validation data set are better (i.e., lower) with the addition of weight assignment to MFRR, as compared with those after *ad hoc* methods.

Table 5.8: Experiment result for Ishibuchi method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.7788	106.5721	228.4726
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	106.4905	227.5584
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	107.1785	228.5785
4	w_{j_1,j_2}^{sum}	1.0000	106.5753	227.6251
5	w_{j_1,j_2}^{max}	1.0000	106.0046	225.9680

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7019	47.1516	106.1116	1.0000	48.7805	101.4804
2	$w_{j_1,j_2}^{\text{data}}$	0.6374	47.1516	153.2022	1.0000	49.6366	98.5076
3	w_{j_1,j_2}^{sum}	0.6374	47.1516	152.1520	1.0000	48.6405	86.3285
4	w_{j_1,j_2}^{max}	0.6717	47.1516	115.5480	1.0000	48.0081	88.6654

(b)

Table 5.8 continued

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.6937	38.6375	88.1845	1.0000	41.8709	72.5531
2	$w_{j_1,j_2}^{\text{data}}$	0.6635	38.6375	145.3049	1.0000	41.5290	63.6706
3	w_{j_1,j_2}^{sum}	0.6635	38.6375	144.4881	1.0000	41.2514	61.9617
4	$w_{j_1,j_2}^{\max}$	0.6772	38.6375	119.5716	1.0000	41.1154	69.6349

(c)

The results for thermal comfort index prediction for 50% noise with WM and Ishibuchi methods using trapezoidal fuzzy set are depicted in Table 5.9 and Table 5.10, respectively. As an example, in Table 5.9 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.7390 and 1.0000, the SSE values with the training data set are 150.0796 and 174.7350, the SSE values with the validation data set are 453.0441 and 201.4897 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing MFRR.

Table 5.9: Experiment result for WM method using trapezoidal fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.8434	190.9195	324.1541
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	201.1283	284.2870
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	198.9695	262.7024
4	w_{j_1,j_2}^{sum}	1.0000	198.8534	264.0111
5	w_{j_1,j_2}^{max}	1.0000	197.3122	279.3304

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8077	149.8539	322.4655	1.0000	175.4409	222.2149
2	$w_{j_1,j_2}^{\text{data}}$	0.7390	150.0796	453.0441	1.0000	174.7350	201.4897
3	w_{j_1,j_2}^{sum}	0.7431	150.0344	442.3587	1.0000	174.5167	204.1509
4	w_{j_1,j_2}^{max}	0.8077	149.8539	322.4655	1.0000	175.4409	222.2149

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7500	137.8756	339.0881	1.0000	165.7298	233.5228
2	$w_{j_1,j_2}^{\text{data}}$	0.7157	139.4584	464.0820	1.0000	162.3399	204.6342
3	w_{j_1,j_2}^{sum}	0.7157	139.4151	461.5385	1.0000	162.0983	208.0283
4	w_{j_1,j_2}^{max}	0.7253	138.2923	375.0784	1.0000	164.4696	238.0886

(c)

Similar results are obtained using the Ishibuchi method for trapezoidal fuzzy set. As an example, in Table 5.10 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.7349 and 1.0000, the SSE values with the training data set are 150.6306 and 175.3762, the SSE values with the validation data set are 380.1142 and 244.7217 after WFRI and MFRR, respectively. Again, the SSE values for both training and validation data set are better (i.e., lower) with the addition of weight assignment to MFRR, as compared with those after *ad hoc* methods.

Table 5.10: Experiment result for Ishibuchi method using trapezoidal fuzzy set (a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.8599	205.8045	287.4676
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	208.8084	277.4734
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	209.3725	277.4239
4	w_{j_1,j_2}^{sum}	1.0000	208.1449	269.8215
5	w_{j_1,j_2}^{max}	1.0000	205.4537	270.4410

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7637	150.6306	338.8724	1.0000	176.6372	263.7589
2	$w_{j_1,j_2}^{\text{data}}$	0.7349	150.6306	380.1142	1.0000	175.3762	244.7217
3	w_{j_1,j_2}^{sum}	0.7390	150.6306	375.8749	1.0000	174.7339	224.9690
4	w_{j_1,j_2}^{max}	0.7637	150.6306	340.9650	1.0000	176.1611	242.8402

(b)

Table 5.10 continued

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7871	137.8069	297.9060	1.0000	165.1292	230.4137
2	$w_{j_1,j_2}^{\text{data}}$	0.7692	137.8069	368.8752	1.0000	161.8734	217.0013
3	w_{j_1,j_2}^{sum}	0.7692	137.8069	371.9134	1.0000	162.3331	208.9218
4	$w_{j_1,j_2}^{\max}$	0.7734	137.8069	336.0141	1.0000	163.9207	237.1158

(c)

The results for thermal comfort index prediction for 50% noise with WM and Ishibuchi methods using triangular fuzzy set are depicted in Table 5.11 and Table 5.12, respectively. As an example, in Table 5.11 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.5398 and 1.0000, the SSE values with the training data set are 145.1988 and 162.2798, the SSE values with the validation data set are 341.3308 and 199.1935 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing MFRR.

Table 5.11: Experiment result for WM method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.5714	165.8439	264.4566
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	169.9771	245.4844
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	181.7546	267.1488
4	w_{j_1,j_2}^{sum}	1.0000	181.5447	265.5801
5	w_{j_1,j_2}^{max}	1.0000	173.4961	267.5340

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.6429	145.1869	260.9858	1.0000	157.9330	215.9117
2	$w_{j_1,j_2}^{\text{data}}$	0.5398	145.1988	341.3308	1.0000	162.2798	199.1935
3	w_{j_1,j_2}^{sum}	0.5398	145.1957	340.0483	1.0000	161.0418	199.8306
4	w_{j_1,j_2}^{max}	0.6085	145.1589	272.4789	1.0000	157.8907	216.9158

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.6250	139.6173	267.8421	1.0000	151.0248	214.5617
2	$w_{j_1,j_2}^{\text{data}}$	0.4945	144.5965	393.0873	1.0000	153.4888	203.0909
3	w_{j_1,j_2}^{sum}	0.4973	144.4097	395.5867	1.0000	151.7925	205.2187
4	w_{j_1,j_2}^{max}	0.5495	141.4161	325.3155	1.0000	150.8532	225.4122

(c)

Similar results are obtained using the Ishibuchi method for triangular fuzzy set. As an example, in Table 5.12 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.5165 and 1.0000, the SSE values with the training data set are 150.7535 and 166.1342, the SSE values with the validation data set are 307.2851 and 245.0525 after WFRI and MFRR, respectively. Again, the SSE values for both training and validation data set are better (i.e., lower) with the addition of weight assignment to MFRR, as compared with those after *ad hoc* methods.

Table 5.12: Experiment result for Ishibuchi method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.7857	227.9901	340.3040
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	229.2374	335.8764
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	228.4928	340.2818
4	w_{j_1,j_2}^{sum}	1.0000	228.2537	335.8226
5	w_{j_1,j_2}^{max}	1.0000	225.8468	326.6365

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.5646	150.7535	271.8332	1.0000	160.5655	252.4810
2	$w_{j_1,j_2}^{\text{data}}$	0.5165	150.7535	307.2851	1.0000	166.1342	245.0525
3	w_{j_1,j_2}^{sum}	0.5165	150.7535	306.3603	1.0000	163.7646	233.0698
4	w_{j_1,j_2}^{max}	0.5495	150.7535	278.6938	1.0000	159.9516	236.7161

(b)

Table 5.12 continued

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.6154	141.3507	247.2929	1.0000	152.4265	212.4788
2	$w_{j_1,j_2}^{\text{data}}$	0.5769	141.3507	301.2397	1.0000	154.5130	207.1479
3	w_{j_1,j_2}^{sum}	0.5769	141.3507	300.6176	1.0000	152.9823	208.2742
4	$w_{j_1,j_2}^{\max}$	0.5948	141.3507	277.5055	1.0000	151.4474	222.2882

(c)

5.3.2 Benchmark Example 2: Auto-Miles per Gallon (Auto-MPG)

The DOM scores of the resulting FIS models are measured using Equation 3.5 (see Chapter 3 Section 3.4), in which $N_1 = N_2 = 93$ is used. The results for Auto-MPG with WM and Ishibuchi methods using trapezoidal fuzzy set are depicted in Table 5.13 and Table 5.14, respectively. As an example, in Table 5.13 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.6758 and 1.0000, the SSE values with the training data set are 1.7115e+03 and 1.7884e+03, the SSE values with the validation data set are 1.0591e+04 and 1.0039e+04 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing MFRR.

Table 5.13: Experiment result for WM method using trapezoidal fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$							
		Without Relabeling			With Relabeling		
	Strategies	DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.9049	2.1079e+03	1.1818e+04	1.0000	2.1058e+03	1.1401e+04
2	$w_{j_1,j_2}^{\text{data}}$	0.7714	2.1005e+03	1.1819e+04	1.0000	2.1224e+03	1.1827e+04
3	w_{j_1,j_2}^{sum}	0.7714	2.1004e+03	1.1819e+04	1.0000	2.1227e+03	1.1832e+04
4	w_{j_1,j_2}^{max}	0.9049	2.1078e+03	1.1818e+04	1.0000	2.1081e+03	1.1483e+04

(a)

For $k_1 = k_2 = 5$							
		Without Relabeling			With Relabeling		
	Strategies	DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7981	1.7035e+03	1.0591e+04	1.0000	2.0456e+03	8.8582e+03
2	$w_{j_1,j_2}^{\text{data}}$	0.6758	1.7115e+03	1.0591e+04	1.0000	1.7884e+03	1.0039e+04
3	w_{j_1,j_2}^{sum}	0.6758	1.7120e+03	1.0591e+04	1.0000	1.7801e+03	1.0002e+04
4	w_{j_1,j_2}^{max}	0.7897	1.7010e+03	1.0591e+04	1.0000	1.9132e+03	9.6664e+03

(b)

For $k_1 = k_2 = 7$							
		Without Relabeling			With Relabeling		
	Strategies	DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7381	1.7181e+03	1.0458e+04	1.0000	2.1397e+03	9.0491e+03
2	$w_{j_1,j_2}^{\text{data}}$	0.6300	1.8066e+03	1.9690e+04	1.0000	1.7906e+03	9.6507e+03
3	w_{j_1,j_2}^{sum}	0.6317	1.8070e+03	1.9691e+04	1.0000	1.7903e+03	9.5349e+03
4	w_{j_1,j_2}^{max}	0.7164	1.7161e+03	1.0527e+04	1.0000	2.1407e+03	9.0405e+03

(c)

Similar results are obtained using the Ishibuchi method for trapezoidal fuzzy set. As an example, in Table 5.14 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.7569 and 1.0000, the SSE values with the training data set are 1.7496e+03 and 1.7852e+03, the SSE values with the validation data set are 1.0540e+04 and 1.0510e+04 after WFRI and MFRR, respectively. Again, the SSE values for both training and validation data set are better (i.e., lower) with the addition of weight assignment to MFRR, as compared with those after *ad hoc* methods.

Table 5.14: Experiment result for Ishibuchi method using trapezoidal fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$							
		Without Relabeling			With Relabeling		
	Strategies	DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	2.1543e+03	1.1995e+04	-	-	-
2	$w_{j_1,j_2}^{\text{data}}$	0.9457	2.1543e+03	1.1995e+04	1.0000	2.1543e+03	1.1995e+04
3	w_{j_1,j_2}^{sum}	0.9366	2.1543e+03	1.1995e+04	1.0000	2.1543e+03	1.1995e+04
4	w_{j_1,j_2}^{max}	1.0000	2.1543e+03	1.1995e+04	-	-	-

(a)

For $k_1 = k_2 = 5$							
		Without Relabeling			With Relabeling		
	Strategies	DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8864	1.7496e+03	1.0540e+04	1.0000	1.8035e+03	9.9195e+03
2	$w_{j_1,j_2}^{\text{data}}$	0.7569	1.7496e+03	1.0540e+04	1.0000	1.7852e+03	1.0510e+04
3	w_{j_1,j_2}^{sum}	0.7569	1.7496e+03	1.0540e+04	1.0000	1.7868e+03	1.0532e+04
4	w_{j_1,j_2}^{max}	0.7843	1.7496e+03	1.0540e+04	1.0000	1.7852e+03	1.0309e+04

(b)

Table 5.14 continued

For $k_1 = k_2 = 7$							
		Without Relabeling			With Relabeling		
	Strategies	DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8156	1.7100e+03	1.0424e+04	1.0000	1.8834e+03	1.0397e+04
2	$w_{j_1,j_2}^{\text{data}}$	0.6755	1.7100e+03	1.0424e+04	1.0000	1.7886e+03	1.0291e+04
3	w_{j_1,j_2}^{sum}	0.6754	1.7100e+03	1.0424e+04	1.0000	1.7936e+03	9.9495e+03
4	w_{j_1,j_2}^{max}	0.7212	1.7100e+03	1.0424e+04	1.0000	1.8665e+03	1.0257e+04

(c)

The results for Auto-MPG with WM and Ishibuchi methods using triangular fuzzy set are depicted in Table 5.15 and Table 5.16, respectively. As an example, in Table 5.15 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.2355 and 1.0000, the SSE values with the training data set are 3.2205e+03 and 1.8414e+03, the SSE values with the validation data set are 2.1815e+04 and 8.7760e+03 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR. Besides that, better (i.e., lower) SSE values are obtained, for both training and validation data sets, after introducing MFRR.

Table 5.15: Experiment result for WM method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.9948	2.1720e+03	1.2749e+04	1.0000	2.1725e+03	1.2749e+04
2	$w_{j_1,j_2}^{\text{data}}$	0.6560	2.1580e+03	1.2743e+04	1.0000	2.1661e+03	1.2745e+04
3	w_{j_1,j_2}^{sum}	0.6505	2.1583e+03	1.2743e+04	1.0000	2.1808e+03	1.2754e+04
4	w_{j_1,j_2}^{max}	0.9156	2.1638e+03	1.2747e+04	1.0000	2.1681e+03	1.2748e+04

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.5040	1.8855e+03	1.0367e+04	1.0000	1.9322e+03	8.9004e+03
2	$w_{j_1,j_2}^{\text{data}}$	0.2355	3.2205e+03	2.1815e+04	1.0000	1.8414e+03	8.7760e+03
3	w_{j_1,j_2}^{sum}	0.2355	3.2303e+03	2.1834e+04	1.0000	1.8373e+03	9.3983e+03
4	w_{j_1,j_2}^{max}	0.4101	2.0140e+03	1.2795e+04	1.0000	1.8505e+03	9.5889e+03

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.4384	1.7513e+03	1.1203e+04	1.0000	2.0467e+03	9.2437e+03
2	$w_{j_1,j_2}^{\text{data}}$	0.3322	1.9620e+03	1.9710e+04	1.0000	1.8285e+03	1.0458e+04
3	w_{j_1,j_2}^{sum}	0.3306	1.9741e+03	1.9797e+04	1.0000	1.8248e+03	1.0433e+04
4	w_{j_1,j_2}^{max}	0.3885	1.7665e+03	1.3745e+04	1.0000	1.9044e+03	1.0024e+04

(c)

Similar results are obtained using the Ishibuchi method for triangular fuzzy set. As an example, in Table 5.16 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.7085 and 1.0000, the SSE values with the training data set are 1.9037+03 and 1.9045+03, the SSE values with the validation data set are 1.1392+04 and 1.1378+04 after WFRI and MFRR, respectively. Again, the SSE values for both training and validation data set are better (i.e., lower) with the addition of weight assignment to MFRR, as compared with those after *ad hoc* methods.

Table 5.16: Experiment result for Ishibuchi method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.9792	2.4121e+03	1.4000e+04
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	2.4070e+03	1.3998e+04
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	2.4122e+03	1.4000e+04
4	w_{j_1,j_2}^{sum}	1.0000	2.4122e+03	1.4000e+04
5	w_{j_1,j_2}^{max}	1.0000	2.4121e+03	1.4000e+04

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.8703	1.9037e+03	1.1392e+04	1.0000	1.8996e+03	1.1347e+04
2	$w_{j_1,j_2}^{\text{data}}$	0.7085	1.9037e+03	1.1392e+04	1.0000	1.9045e+03	1.1378e+04
3	w_{j_1,j_2}^{sum}	0.7030	1.9037e+03	1.1392e+04	1.0000	1.9045e+03	1.1391e+04
4	w_{j_1,j_2}^{max}	0.7169	1.9037e+03	1.1392e+04	1.0000	1.9039e+03	1.1383e+04

(b)

Table 5.16 continued

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.6733	1.7929e+003	1.0785e+004	1.0000	1.8375e+003	1.0350e+004
2	$w_{j_1,j_2}^{\text{data}}$	0.4414	1.7929e+003	1.0783e+004	1.0000	1.8232e+003	1.0741e+004
3	w_{j_1,j_2}^{sum}	0.4409	1.7929e+003	1.0783e+004	1.0000	1.8246e+003	1.0758e+004
4	$w_{j_1,j_2}^{\max}$	0.4512	1.7929e+003	1.0784e+004	1.0000	1.8223e+003	1.0726e+004

(c)

5.3.3 Benchmark Example 3: Mercury-in-Fish

The DOM scores of the resulting FIS models are measured using Equation 3.5 (see Chapter 3 Section 3.4), in which $N_1 = N_2 = 12$ is used. The results for mercury-in-Fish with WM and Ishibuchi methods using trapezoidal fuzzy set are depicted in Table 5.17 and Table 5.18, respectively. As an example, in Table 5.17 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM are 0.5893 and 1.0000, the SSE values with the training data set are 0.4639 and 0.5238, the SSE values with the validation data set are 0.3020 and 0.3046 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR.

Table 5.17: Experiment result for WM method using trapezoidal fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.7143	0.6337	0.3202	1.0000	0.6670	0.3143
2	$w_{j_1,j_2}^{\text{data}}$	0.6964	0.6228	0.3165	1.0000	0.6547	0.3080
3	w_{j_1,j_2}^{sum}	0.6964	0.6228	0.3165	1.0000	0.6604	0.3069
4	$w_{j_1,j_2}^{\max}$	0.7143	0.6337	0.3202	1.0000	0.6592	0.3130

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.5417	0.4539	0.3061	1.0000	0.6091	0.3872
2	$w_{j_1,j_2}^{\text{data}}$	0.5893	0.4639	0.3020	1.0000	0.5238	0.3046
3	w_{j_1,j_2}^{sum}	0.6012	0.4641	0.3020	1.0000	0.5243	0.3045
4	$w_{j_1,j_2}^{\max}$	0.5536	0.4541	0.3060	1.0000	0.5147	0.2999

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.4048	0.4609	0.2863	1.0000	0.6285	0.4446
2	$w_{j_1,j_2}^{\text{data}}$	0.3393	0.4659	0.2883	1.0000	0.5296	0.3118
3	w_{j_1,j_2}^{sum}	0.3571	0.4655	0.2882	1.0000	0.5299	0.3061
4	$w_{j_1,j_2}^{\max}$	0.4048	0.4610	0.2860	1.0000	0.5440	0.2976

(c)

Similar results are obtained using the Ishibuchi method for trapezoidal fuzzy set. As an example, in Table 5.18 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.5476 and 1.0000, the SSE values with the training data set are 0.4509 and 0.5481, the SSE values with the validation data set are 0.3095 and 0.3137 after WFRI and MFRR, respectively. Again, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR.

Table 5.18: Experiment result for Ishibuchi method using trapezoidal fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.7857	0.6339	0.3095
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	0.6578	0.3052
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	0.6529	0.3061
4	w_{j_1,j_2}^{sum}	1.0000	0.6528	0.3061
5	w_{j_1,j_2}^{max}	1.0000	0.6480	0.3060

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.5536	0.4509	0.3131	1.0000	0.5159	0.3041
2	$w_{j_1,j_2}^{\text{data}}$	0.5476	0.4509	0.3095	1.0000	0.5481	0.3137
3	w_{j_1,j_2}^{sum}	0.5476	0.4509	0.3096	1.0000	0.5477	0.3122
4	w_{j_1,j_2}^{max}	0.5714	0.4509	0.3130	1.0000	0.5414	0.2998

(b)

Table 5.18 continued

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.3929	0.4549	0.2818	1.0000	0.6284	0.4349
2	$w_{j_1,j_2}^{\text{data}}$	0.3750	0.4549	0.2818	1.0000	0.5644	0.3159
3	w_{j_1,j_2}^{sum}	0.4048	0.4549	0.2818	1.0000	0.5246	0.3037
4	w_{j_1,j_2}^{max}	0.4524	0.4549	0.2818	1.0000	0.6285	0.4446

(c)

The DOM scores of the resulting FIS models are measured using Equation 3.5 (see Chapter 3 Section 3.4), in which $N_1 = N_2 = 12$ is used. The results for mercury-in-Fish with WM and Ishibuchi methods using triangular fuzzy set are depicted in Table 5.19 and Table 5.20, respectively. As an example, in Table 5.19 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM are 0.3452 and 1.0000, the SSE values with the training data set are 0.4712 and 0.5298, the SSE values with the validation data set are 0.3002 and 0.3040 after WFRI and MFRR, respectively. In short, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR.

Table 5.19: Experiment result for WM method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.5357	0.6196	0.3394	1.0000	0.6525	0.3287
2	$w_{j_1,j_2}^{\text{data}}$	0.5357	0.6049	0.3232	1.0000	0.6357	0.3365
3	w_{j_1,j_2}^{sum}	0.5357	0.6049	0.3232	1.0000	0.6352	0.3386
4	$w_{j_1,j_2}^{\max}$	0.6071	0.6192	0.3390	1.0000	0.6367	0.3257

(a)

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.4167	0.4744	0.3041	1.0000	0.5168	0.2978
2	$w_{j_1,j_2}^{\text{data}}$	0.3452	0.4712	0.3002	1.0000	0.5298	0.3040
3	w_{j_1,j_2}^{sum}	0.3512	0.4712	0.3002	1.0000	0.5350	0.3158
4	$w_{j_1,j_2}^{\max}$	0.5060	0.4728	0.3032	1.0000	0.5168	0.2978

(b)

Table 5.19 continued

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.4821	0.4742	0.2859	1.0000	0.5468	0.2994
2	$w_{j_1,j_2}^{\text{data}}$	0.4405	0.4785	0.2856	1.0000	0.5324	0.2982
3	w_{j_1,j_2}^{sum}	0.4405	0.4783	0.2856	1.0000	0.5076	0.2976
4	w_{j_1,j_2}^{max}	0.4940	0.4736	0.2855	1.0000	0.5195	0.3104

(c)

Similar results are obtained using the Ishibuchi method for triangular fuzzy set. As an example, in Table 5.20 (b), for $w_{j_1,j_2}^{\text{data}}$, the DOM scores are 0.5536 and 1.0000, the SSE values with the training data set are 0.4854 and 0.5408, the SSE values with the validation data set are 0.2968 and 0.2981 after WFRI and MFRR, respectively. Again, the monotonicity property of the zero-order TSK FIS model is fulfilled after MFRR.

Table 5.20: Experiment result for Ishibuchi method using triangular fuzzy set a) For $k_1 = k_2 = 3$, (b) For $k_1 = k_2 = 5$ and (c) For $k_1 = k_2 = 7$

For $k_1 = k_2 = 3$				
	Strategies	DOM	SSE with training set	SSE with validation set
1	Without Relabeling	0.7143	0.6392	0.3194
2	$w_{j_1,j_2}^{\text{One-Zero}}$	1.0000	0.6496	0.3163
3	$w_{j_1,j_2}^{\text{data}}$	1.0000	0.6455	0.3173
4	w_{j_1,j_2}^{sum}	1.0000	0.6454	0.3210
5	w_{j_1,j_2}^{max}	1.0000	0.6436	0.3180

(a)

Table 5.20 continued

For $k_1 = k_2 = 5$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.4643	0.4854	0.2968	1.0000	0.5593	0.3031
2	$w_{j_1,j_2}^{\text{data}}$	0.5536	0.4854	0.2968	1.0000	0.5408	0.2981
3	w_{j_1,j_2}^{sum}	0.5536	0.4854	0.2968	1.0000	0.5415	0.3016
4	w_{j_1,j_2}^{max}	0.5476	0.4854	0.2968	1.0000	0.5282	0.3081

(b)

For $k_1 = k_2 = 7$							
	Strategies	Without Relabeling			With Relabeling		
		DOM	SSE with training set	SSE with validation set	DOM	SSE with training set	SSE with validation set
1	$w_{j_1,j_2}^{\text{One-Zero}}$	0.4286	0.4662	0.2875	1.0000	0.5105	0.2965
2	$w_{j_1,j_2}^{\text{data}}$	0.4464	0.4662	0.2875	1.0000	0.5316	0.2925
3	w_{j_1,j_2}^{sum}	0.4524	0.4662	0.2875	1.0000	0.5220	0.3076
4	w_{j_1,j_2}^{max}	0.5000	0.4662	0.2875	1.0000	0.5054	0.2933

(c)

5.3.4 Remarks

In short, these experimental results indicate that while the training and validation data sets are *non-monotone* (noisy), by including Stages 3 to 5, a better DOM score can be obtained. Interestingly, some SSE values for the validation data set are better with the addition of Stages 3 to 5, as compared with those from Stage 2. The outcome ascertains that the monotonicity property and Proposition 1 are important additional information for developing *monotone* FIS models.

5.4 Summary

In this chapter, a series of benchmark examples are used for evaluation to demonstrate the usefulness of our proposed framework for modeling *monotone* TSK FIS models with *ad hoc* methods. Positive results, i.e., a perfect DOM value (always equal to 1) and a better (i.e., lower) SSE value were obtained. The results were discussed thoroughly.

In the next chapter, an analysis of the system identification method for single and multi-attribute *monotone* data sets will be presented.

CHAPTER 6

SYSTEM IDENTIFICATION METHOD WITH MONOTONE DATA

6.1 Chapter Overview

In the previous chapter, an application of the six-stage framework to the prediction of the thermal comfort index and the two real-world benchmark problems was presented. In this chapter, the use of a system identification-based framework to develop *monotone* fuzzy If-Then rules using multi-attribute data for modelling a zero-order Takagi-Sugeno-Kang (TSK) Fuzzy Inference System (FIS) is analysed. The analysis for a system identification method is conducted to ascertain whether the system identification method could always produce *monotone* fuzzy rules with multi-attribute *monotone* data sets. However the empirical studies indicate that fuzzy If-Then rules generated using the system identification method : (1) satisfy the monotonicity property when a single-attribute *monotone* data set is considered; (2) do not always satisfy the monotonicity property when a multi-attribute *monotone* data set is considered. As such, two formulations, i.e., (1) based on a *monotone fuzzy rules relabeling* (MFRR) method; and (2) based on a constrained derivative-based optimization method, are devised. The results demonstrate that our proposed framework produces better accuracy (i.e., a lower sum square error), and DOM score. A series of benchmark problems are used for evaluation to demonstrate the usefulness of our proposed system identification-based framework for developing *monotone* TSK FIS models with system identification method.

In Section 6.2, the background of the system identification method is introduced. In Section 6.3, an analysis of *monotone* TSK FIS models with a strong fuzzy partition strategy is presented. In Section 6.4, two solutions for practical modeling of *monotone*

TSK FIS models are devised. In Section 6.5, a series of experiments using benchmark problems are conducted. Finally, concluding remarks are given in Section 6.6.

6.2 System Identification Method

A schematic diagram of parameter identification with consideration of $(\boldsymbol{x}^p; y^p)$ is illustrated in Figure 6.1.

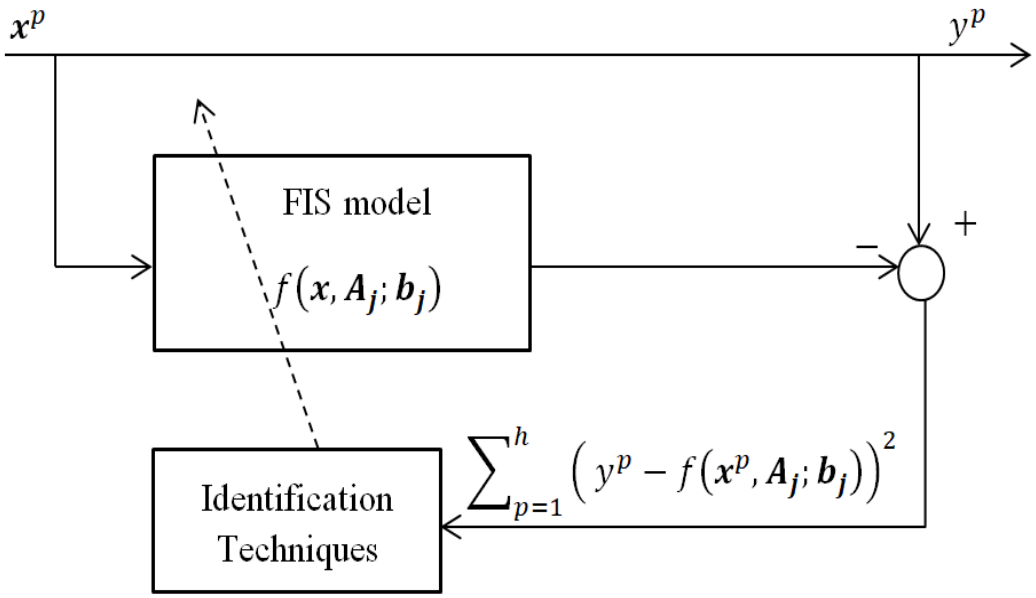

Figure 6.1: A block diagram of parameter identification

A system identification method for an FIS model can be summarized into 4 steps (Jang et al., 1997), as follows.

Step 1: Specify and parameterize an FIS model, i.e., $f(\boldsymbol{x}, \boldsymbol{A_j}; \boldsymbol{b_j})$, that represents a system to be identified. Then, $\boldsymbol{A_j} = (A_{1,\cdots,1}, A_{1,\cdots,2}, \cdots, A_{k_1,\cdots,k_n})$ in the input space and $\boldsymbol{b_j} = (b_{1,\cdots,1}, b_{1,\cdots,2}, \cdots, b_{k_1,\cdots,k_n})$ need to be identified. A strong partition is used.

Step 2: Perform parameter identification to form a set of fuzzy If-Then rules, $[\boldsymbol{A_j}, \boldsymbol{b_j}] = [\boldsymbol{A_j^*}, \boldsymbol{b_j^*}]$ that best fits the data set, i.e., $(\boldsymbol{x}^p; y^p)$.

Step 3: Conduct a validation test to see if the model has been identified correctly for the problem under scrutiny in accordance with the test data set.

Step 4: Terminate the procedure once the validation test is satisfactory. Otherwise, go back to Step 1.

6.3 Identification of Fuzzy Rules with Strong Fuzzy Partition Strategy

6.3.1 Unconstrained derivative-based optimization approach

In this section, $\boldsymbol{A_j}$ is pre-fixed. In Step 2, an optimization algorithm is used to search for a set of $\boldsymbol{b_j} = \boldsymbol{b_j^*}$ such that the resulting FIS model, $f(\boldsymbol{x}, \boldsymbol{A_j}; \boldsymbol{b_j^*})$, best represents the model, e.g., producing the lowest Sum Square Error (SSE) with respect to a training data set i.e., $(\boldsymbol{x}^p; y^p)$, as follows.

$$\mathrm{SSE}(\boldsymbol{b_j}) = \sum_{p=1}^{h} \left(y^p - f(\boldsymbol{x}^p, \boldsymbol{A_j}; \boldsymbol{b_j}) \right)^2 \tag{6.1}$$

The derivative of Equation 6.1 with respect to $b_{1,\cdots,k_n}$ is estimated using the central finite difference method (Nocedal & Wright, 2006), as in Equation 6.2.

$$\frac{\partial \mathrm{SSE}}{\partial b_{1,\cdots,k_n}} = \frac{\mathrm{SSE}(\boldsymbol{b_j^1}) - \mathrm{SSE}(\boldsymbol{b_j^2})}{2\Delta} \tag{6.2}$$

where $\boldsymbol{b_j^1} = \left(b_{1,\cdots,1}, \cdots b_{1,\cdots,k_n} + \Delta, \cdots, b_{k_1,\cdots,k_n}\right)$; $\boldsymbol{b_j^2} = \left(b_{1,\cdots,1}, \cdots b_{1,\cdots,k_n} - \Delta, \cdots, b_{k_1,\cdots,k_n}\right)$ and Δ is a pre-specified limit.

6.3.2 Simulated Example 4.1: Single Attribute *Monotone* data

The generated fuzzy If-Then rules using system identification method based on the simulated Example 4.1 for (a) $k_1 = 3$, (b) $k_1 = 5$ and (c) $k_1 = 7$ using trapezoidal and triangular fuzzy sets are illustrated in Table 6.1 and Table 6.2, respectively. *Monotone* fuzzy If-Then rules are produced using system identification method with trapezoidal and triangular fuzzy sets.

Table 6.1: Generated fuzzy If-Then rules using trapezoidal fuzzy set a) For $k_1 = 3$, (b) For $k_1 = 5$ and (c) For $k_1 = 7$

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$
0.0559	0.2324	0.3566

(a)

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	$A_{1,5}$
0.0307	0.1351	0.2316	0.3101	0.3676

(b)

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	$A_{1,5}$	$A_{1,6}$	$A_{1,7}$
0.0208	0.0950	0.1669	0.2313	0.2874	0.3342	0.3720

(c)

Table 6.2: Generated fuzzy If-Then rules using triangular fuzzy set a) For $k_1 = 3$, (b) For $k_1 = 5$ and (c) For $k_1 = 7$

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$
0.0031	0.2378	0.3887

(a)

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	$A_{1,5}$
0.0004	0.1235	0.2330	0.3196	0.3824

(b)

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	$A_{1,5}$	$A_{1,6}$	$A_{1,7}$
0.0002	0.0830	0.1614	0.2319	0.2924	0.3420	0.3815

(c)

6.3.3 Simulated Example 4.2: Multi-attribute *Monotone* data

The generated fuzzy If-Then rules using system identification method based on the simulated Example 4.2 for $k_1 = k_1 = 3$ using trapezoidal and triangular fuzzy sets are illustrated in Table 6.3 and Table 6.4, respectively. *Monotone* fuzzy If-Then rules are produced using system identification method with trapezoidal and triangular fuzzy sets.

Table 6.3: Generated fuzzy If-Then rules for $k_1 = k_1 = 3$ using trapezoidal fuzzy set

$A_{1,3}$	0.5034	0.7487	0.9620
$A_{1,2}$	0.2671	0.5607	0.7254
$A_{1,1}$	0.0334	0.2694	0.5085
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$

Table 6.4: Generated fuzzy If-Then rules for $k_1 = k_1 = 3$ using triangular fuzzy set

$A_{1,3}$	0.4870	0.7911	1.0652
$A_{1,2}$	0.2459	0.5636	0.7624
$A_{1,1}$	-0.1017	0.2487	0.5036
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$

The generated fuzzy If-Then rules using system identification method based on the simulated Example 4.2 for $k_1 = k_2 = 5$ using trapezoidal and triangular fuzzy set are illustrated in Table 6.5 and Table 6.6, respectively. *Non-monotone* and *complete* fuzzy If-Then rules are produced using system identification method for trapezoidal and triangular fuzzy set, respectively. As an example, in Table 6.5, for $A_{j_1=2,j_2=2}$, $b_{2,2} = 0.4301$, and for $A_{j_1=2,j_2=3}$, $b_{2,3} = 0.3848$. Since $b_{2,3} < b_{2,2}$, *non-monotone* fuzzy If-Then rules are obtained.

Table 6.5: Generated fuzzy If-Then rules for $k_1 = k_1 = 5$ using trapezoidal fuzzy set

$A_{1,5}$	0.4999	0.5502	0.7357	0.9227	0.9712
$A_{1,4}$	0.4530	0.4661	0.6848	0.8609	0.9423
$A_{1,3}$	**0.2616**	**-0.2017**	0.5479	0.7067	0.7290
$A_{1,2}$	0.0763	**0.4301**	**0.3848**	0.4891	0.5320
$A_{1,1}$	0.0311	0.0667	0.2633	0.4512	0.5014
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

Table 6.6: Generated fuzzy If-Then rules for $k_1 = k_1 = 5$ using triangular fuzzy set

$A_{1,5}$	0.4997	0.5303	0.7497	0.9512	0.9818
$A_{1,4}$	0.4702	0.5027	0.7074	0.9091	0.9456
$A_{1,3}$	0.2593	0.2686	0.5091	**0.7423**	**0.7255**
$A_{1,2}$	0.0486	0.0683	0.2691	0.5039	0.5365
$A_{1,1}$	0.0230	0.0480	0.2592	0.4699	0.4978
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

The generated fuzzy If-Then rules using the system identification method based on the simulated Example 4.2, for $k_1 = k_2 = 7$ using trapezoidal and triangular fuzzy set are illustrated in Table 6.7 and Table 6.8, respectively. *Non-monotone* and *complete* fuzzy If-Then rules are produced using system identification method for trapezoidal and triangular fuzzy set, respectively. As an example, in Table 6.7, for $A_{j_1=6,j_2=3}$, $b_{6,3} = 0.6215$, and for $A_{j_1=7,j_2=3}$, $b_{7,3} = 0.5738$. Since $b_{7,3} < b_{6,3}$, *non-monotone* fuzzy If-Then rules are obtained.

Table 6.7: Generated fuzzy If-Then rules for $k_1 = k_1 = 7$ using trapezoidal fuzzy set

$A_{1,7}$	0.4999	0.5290	**0.5738**	0.7410	0.8993	0.9502	0.9789
$A_{1,6}$	0.4728	0.4997	**0.6215**	0.7065	0.8451	0.9214	0.9492
$A_{1,5}$	0.3998	0.4538	0.5367	0.6661	0.7742	**1.3183**	**0.2511**
$A_{1,4}$	0.2600	0.2844	**0.5292**	**0.2856**	**0.4866**	**0.4082**	0.8173
$A_{1,3}$	0.1127	0.1450	**0.1506**	**0.7333**	**0.5014**	**0.5338**	0.6080
$A_{1,2}$	0.0474	0.0891	**0.2245**	0.2776	0.4535	0.4999	0.5297
$A_{1,1}$	0.0231	0.0483	0.1126	0.2595	0.3997	0.4724	0.4999
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

Table 6.8: Generated fuzzy If-Then rules for $k_1 = k_1 = 7$ using triangular fuzzy set

$A_{1,7}$	0.4999	0.5158	0.5842	0.7494	0.9002	0.9614	0.9797
$A_{1,6}$	0.4843	0.5012	0.5674	0.7229	0.8745	0.9536	0.9646
$A_{1,5}$	0.4152	0.4333	0.4865	0.6576	0.8063	0.8408	0.8938
$A_{1,4}$	0.2574	**0.2764**	**0.2296**	**0.4937**	0.6649	**0.7926**	**0.7015**
$A_{1,3}$	0.0993	**0.1166**	**0.1098**	**0.5689**	**0.5081**	0.5432	0.6140
$A_{1,2}$	0.0381	0.0545	0.1021	0.2783	0.4298	0.5050	0.5118
$A_{1,1}$	0.0183	0.0383	0.1003	0.2571	0.4146	0.4841	0.5011
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

6.4 Identification of *Monotone* Fuzzy If-Then Rules

Motivated by the findings in Section 6.3.3, two solutions for practical modelling of data-based *monotone* TSK FIS with system identification are proposed. As depicted in Figure 6.2, **Solution 1** consists of four stages, i.e., (A), (B), (C), and (D), with a fuzzy rule re-labelling procedure, while **Solution 2** consists of three stages, i.e., (A), (E), and (D). For clarity, the training data and the fuzzy partition design from the simulated Example 4.2 are used for explanation, as follows.

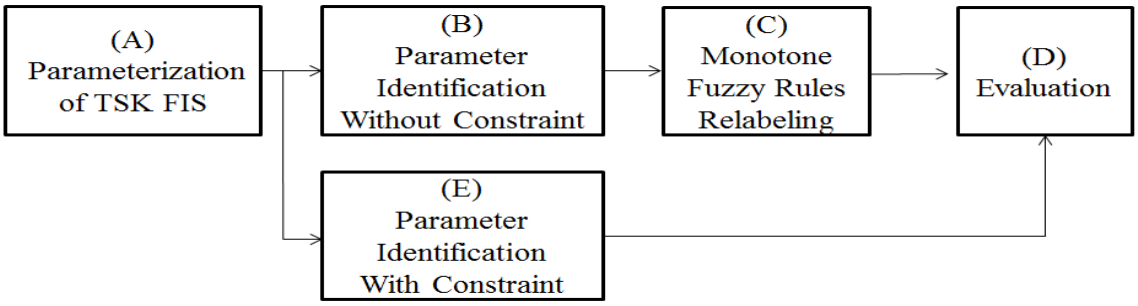

Figure 6.2: The proposed system identification-based framework for *monotone* TSK FIS modelling

6.4.1 Solution 1

In Stage (A), the parameters of the TSK FIS model, i.e., $f(\boldsymbol{x}, \boldsymbol{A_j}; \boldsymbol{b_j})$, are specified (see Step 1 in Section 6.2), in which $\boldsymbol{A_j}$ is pre-specified. In Stage (B), a parameter identification method is used (see Step 2 in Section 6.2), and $\boldsymbol{b_j}$ is identified. In Stages (C), an MFRR (see Stage 5 in Section 4.4) is used to re-label the *non-monotone* fuzzy rules obtained from Stage (B). In Stage (D), the accuracy of the generated FIS model using system identification method is validated by using the Sum Square Error (SSE), i.e., Equation 4.11.

6.4.2 Simulation Studies with Simulated Example 4.2

As depicted in Table 6.9 and Table 6.10, *complete and monotone* fuzzy If-Then rules are produced after Stage (C), for $k_1 = k_2 = 5$ using trapezoidal and triangular fuzzy set, respectively.

Table 6.9: Fuzzy rules for $k_1 = k_2 = 5$ after Stage (C) for trapezoidal fuzzy set

$A_{1,5}$	0.4999	0.5502	0.7357	0.9227	0.9712
$A_{1,4}$	0.4530	0.4661	0.6848	0.8609	0.9423
$A_{1,3}$	0.1633	0.1633	0.5479	0.7067	0.7290
$A_{1,2}$	0.0763	0.1633	0.3848	0.4891	0.5320
$A_{1,1}$	0.0311	0.0667	0.2633	0.4512	0.5014
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

Table 6.10: Fuzzy rules for $k_1 = k_2 = 5$ after Stage (C) for triangular fuzzy set

$A_{1,5}$	0.4997	0.5303	0.7497	0.9512	0.9818
$A_{1,4}$	0.4702	0.5027	0.7074	0.9091	0.9456
$A_{1,3}$	0.2593	0.2686	0.5091	0.7339	0.7339
$A_{1,2}$	0.0486	0.0683	0.2691	0.5039	0.5365
$A_{1,1}$	0.0230	0.0480	0.2592	0.4699	0.4978
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

Again, as depicted in Table 6.11 and Table 6.12, *complete and monotone* fuzzy If-Then rules are produced after Stage (C), for $k_1 = k_2 = 7$ using trapezoidal and triangular fuzzy set, respectively.

Table 6.11: Fuzzy rules for $k_1 = k_2 = 7$ after Stage (C) for trapezoidal fuzzy set

$A_{1,7}$	0.5000	0.5290	0.5980	0.7410	0.8990	0.9500	0.9790
$A_{1,6}$	0.4730	0.5000	0.5980	0.7070	0.8450	0.9210	0.9490
$A_{1,5}$	0.4000	0.4540	0.5370	0.6660	0.7740	0.7950	0.7950
$A_{1,4}$	0.2600	0.2840	0.4970	0.4970	0.4970	0.4970	0.7950
$A_{1,3}$	0.1130	0.1450	0.1870	0.4970	0.4970	0.4970	0.6080
$A_{1,2}$	0.0480	0.0890	0.1870	0.2770	0.4540	0.4970	0.5300
$A_{1,1}$	0.0230	0.0480	0.1120	0.2590	0.3990	0.4730	0.5000
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

Table 6.12: Fuzzy rules for $k_1 = k_2 = 7$ after Stage (C) for triangular fuzzy set

$A_{1,7}$	0.5000	0.5160	0.5840	0.7500	0.9000	0.9620	0.9800
$A_{1,6}$	0.4840	0.5010	0.5680	0.7230	0.8750	0.9530	0.9640
$A_{1,5}$	0.4150	0.4330	0.4870	0.6570	0.8060	0.8410	0.8940
$A_{1,4}$	0.2550	0.2550	0.2550	0.5230	0.6650	0.7470	0.7470
$A_{1,3}$	0.0990	0.1130	0.1130	0.5230	0.5230	0.5430	0.6140
$A_{1,2}$	0.0380	0.0550	0.1020	0.2780	0.4300	0.5050	0.5120
$A_{1,1}$	0.0180	0.0380	0.1000	0.2570	0.4140	0.4840	0.5010
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

6.4.3 Solution 2

In Stage (A), the parameters of the TSK FIS model, i.e., $f(\boldsymbol{x}, \boldsymbol{A_j}; \boldsymbol{b_j})$, are specified (see Step 1 in Section 6.2). In Stage (E), a parameter identification method with constraints is used. A quadratic augmented objective function i.e., $O(\boldsymbol{b_j}, c)$, to be optimized, is formulated, as follows.

$$O(\boldsymbol{b_j}, c) = \text{SSE}(\boldsymbol{b_j}) + cP(\boldsymbol{b_j}) \tag{6.3}$$

where $\text{SSE}(\boldsymbol{b_j})$ can be obtained from Equation 6.1; c is the variable penalty parameter, and $P(\boldsymbol{b_j})$ is the *penalty function*, as in Equation 6.4.

$$P(\boldsymbol{b_j}) = \sum_{lo=1}^{L} g_{lo}^{+}(\boldsymbol{b_j}) \tag{6.4}$$

where $g_{lo}^{+}(\boldsymbol{b_j}) = \left(max\{0, g_{lo}(\boldsymbol{b_j})\}\right)^2$, and $g_{lo}(\boldsymbol{b_j}) = -\left(b_{\boldsymbol{j}+\boldsymbol{e}_i} - b_{\boldsymbol{j}}\right) \leq 0$, where $lo = 1,2,\cdots,L$.

Algorithm 2 is devised for Stage (E). Again, the DFP method (Davidon, 1991; Fletcher & Powell, 1963) is used to solve $\boldsymbol{b_j}^{(\mathrm{K}+1)}$ (Step 3 of Algorithm 2), with respect to a variable penalty parameter, i.e., $c^{(\mathrm{K})}$, at the K-th iteration. Then, $c^{(\mathrm{K}+1)}$ is updated using $c^{(\mathrm{K}+1)} = \eta c^{(\mathrm{K})}$, where η is a growth parameter, until the convergence criterion (Step 4 of Algorithm 2) is satisfied. The partial derivatives of O are estimated by using the central finite difference method, i.e.,

$$\frac{\partial O}{\partial b_{1,\cdots,k_n}^{(\mathrm{K})}} = \frac{O\left(\boldsymbol{b_j}^{(\mathrm{K}),1}, c^{(\mathrm{K})}\right) - O\left(\boldsymbol{b_j}^{(\mathrm{K}),2}, c^{(\mathrm{K})}\right)}{2\Delta} \tag{6.5}$$

where $\boldsymbol{b_j}^{(\mathrm{K}),1} = \left(b_{1,\cdots,1}^{(\mathrm{K})}, \cdots b_{1,\cdots,k_n}^{(\mathrm{K})} + \Delta, \cdots, b_{k_1,\cdots,k_n}^{(\mathrm{K})}\right)$; $\boldsymbol{b_j}^{(\mathrm{K}),2} = \left(b_{1,\cdots,1}^{(\mathrm{K})}, \cdots b_{1,\cdots,k_n}^{(\mathrm{K})} - \Delta, \cdots, b_{k_1,\cdots,k_n}^{(\mathrm{K})}\right)$. Note that Δ is the pre-specified limit for Equation 6.5 which, together with ε_1 and ε_2, need to be pre-specified. An initial search interval for the direct search method, i.e., $\left[\underline{\beta}, \overline{\beta}\right]$, the growth parameter, i.e., η, and the initial penalty parameter, i.e., $c^{(1)}$, are pre-specified. A golden ratio (Kaufman, 1998), $\tau = (\sqrt{5} - 1)/2 = 0.618$, is used as part of the direct search method in Step 3 of Algorithm 2.

Algorithm 2: Constrained derivative-based optimization-based method

Input: Determine the boundary of $\boldsymbol{b_j}$ and randomly generated $\boldsymbol{b_j}^{(1)}$

Output: $\boldsymbol{b_j^*} = \boldsymbol{b_j}^{(\mathrm{K})}$

1. Set K = 1, initialize η, $c^{(1)}$, $\boldsymbol{G}^{(1)} = \boldsymbol{I}$, $\varepsilon_1, \varepsilon_2$, Δ, $\left[\underline{\beta}, \overline{\beta}\right]$
2. **repeat**
3. $\boldsymbol{b_j}^{(\mathrm{K}+1)} \in \arg\min_{\boldsymbol{b_j}^{(\mathrm{K})}} O\left(\boldsymbol{b_j}^{(\mathrm{K})}, c^{(\mathrm{K})}\right)$
4. $c^{(\mathrm{K}+1)} = \eta c^{(\mathrm{K})}$
5. $\mathrm{K} \leftarrow \mathrm{K} + 1$
6. **until** $\left|O\left(\boldsymbol{b_j}^{(\mathrm{K})}, c^{(\mathrm{K})}\right) - O\left(\boldsymbol{b_j}^{(\mathrm{K}-1)}, c^{(\mathrm{K}-1)}\right)\right| < \varepsilon_1$

In Stages (D), accuracy of the FIS model is measured using the SSE, as in Equation 4.11. In this evaluation, the data set used is denoted as $(\boldsymbol{x}^v; y^v), v = 1,2,\cdots,V$. Again, note that $(\boldsymbol{x}^v; y^v)$ can be a training data set or a validation data set.

An analysis for Stage (E) is summarized in Remark 6.1.

Remark 6.1*:* An analysis of Algorithm 2 (i.e., Stage (E)) is as follows.

6.1.1: Some useful properties of Algorithm 2 pertaining to the penalty conditions are as follows:

P1: $O\left(\boldsymbol{b_j}^{(\mathrm{K})}, c^{(\mathrm{K})}\right) \leq O\left(\boldsymbol{b_j}^{(\mathrm{K}+1)}, c^{(\mathrm{K}+1)}\right)$ is always true, for $\mathrm{K} = 1,2,\cdots,\infty$.

P2: $P\left(\boldsymbol{b_j}^{(\mathrm{K})}\right) \geq P\left(\boldsymbol{b_j}^{(\mathrm{K}+1)}\right)$ is always true, for $\mathrm{K} = 1,2,\cdots,\infty$.

P3: $\mathrm{SSE}\left(\boldsymbol{b_j}^{(\mathrm{K})}\right) \leq \mathrm{SSE}\left(\boldsymbol{b_j}^{(\mathrm{K}+1)}\right)$ is always true.

P4: $\mathrm{SSE}\left(\boldsymbol{b_j^*}\right) \geq O\left(\boldsymbol{b_j}^{(\mathrm{K})}, c^{(\mathrm{K})}\right) \geq \mathrm{SSE}\left(\boldsymbol{b_j}^{(\mathrm{K})}\right)$ is always true (Chong; & Zak, 2004).

P5: $c^{(\mathrm{K})} < c^{(\mathrm{K}+1)}$ is always true.

6.1.2: Suppose $O(\boldsymbol{b_j}, c)$ is continuous, and if $c^{(K)} \to \infty$ as $K \to \infty$, then the limit of any convergence sub-sequence of $\left\{\boldsymbol{b_j}^{(K)}\right\}$ is a local optimal solution, to an arbitrary precision (Chong; & Zak, 2004).

6.4.4 Simulation Studies with Simulated Example 4.2

As depicted in Table 6.13 and Table 6.14, *complete and monotone* fuzzy If-Then rules are produced after Stage (E), for $k_1 = k_2 = 5$ using trapezoidal and triangular fuzzy set, respectively.

Table 6.13: Fuzzy rules for $k_1 = k_2 = 5$ after Stage (E) for trapezoidal fuzzy set

$A_{1,5}$	0.4990	0.5499	0.7350	0.9243	0.9703
$A_{1,4}$	0.4532	0.4703	0.6914	0.8599	0.9422
$A_{1,3}$	0.2570	0.2614	0.4865	0.7072	0.7446
$A_{1,2}$	0.0753	0.0760	0.4488	0.4890	0.5290
$A_{1,1}$	0.0303	0.0683	0.2613	0.4519	0.5011
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

Table 6.14: Fuzzy rules for $k_1 = k_2 = 5$ after Stage (E) for triangular fuzzy set

$A_{1,5}$	0.4991	0.5305	0.7498	0.9508	0.9818
$A_{1,4}$	0.4710	0.5022	0.7070	0.9107	0.9453
$A_{1,3}$	0.2594	0.2686	0.5113	0.7333	0.7333
$A_{1,2}$	0.0483	0.0691	0.2672	0.5058	0.5346
$A_{1,1}$	0.0233	0.0478	0.2597	0.4692	0.4985
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$

Again, as depicted in Table 6.15 and Table 6.16, *complete and monotone* fuzzy If-Then rules are produced after Stage (E), for $k_1 = k_2 = 7$ using trapezoidal and triangular fuzzy set, respectively.

Table 6.15: Fuzzy rules for $k_1 = k_2 = 7$ after Stage (E) for trapezoidal fuzzy set

$A_{1,7}$	0.4991	0.5281	0.5851	0.7405	0.8987	0.9498	0.9785
$A_{1,6}$	0.4724	0.5003	0.5630	0.7086	0.8448	0.9212	0.9484
$A_{1,5}$	0.3994	0.4536	0.5562	0.6385	0.8026	0.8049	0.8121
$A_{1,4}$	0.2569	0.3022	0.4802	0.4802	0.5013	0.6929	0.7077
$A_{1,3}$	0.1159	0.1262	0.2188	0.4758	0.4879	0.6156	0.6364
$A_{1,2}$	0.0458	0.0872	0.2154	0.2663	0.4541	0.5018	0.5296
$A_{1,1}$	0.0270	0.0489	0.1120	0.2606	0.3990	0.4714	0.4992
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

Table 6.16: Fuzzy rules for $k_1 = k_2 = 7$ after Stage (E) for triangular fuzzy set

$A_{1,7}$	0.4968	0.5170	0.5829	0.7440	0.8994	0.9642	0.9817
$A_{1,6}$	0.4878	0.4987	0.5706	0.7249	0.8785	0.9416	0.9556
$A_{1,5}$	0.4140	0.4348	0.4860	0.6587	0.7948	0.9001	0.9544
$A_{1,4}$	0.2560	0.2798	0.3281	0.4930	0.6863	0.6918	0.7048
$A_{1,3}$	0.1042	0.1055	0.2079	0.3221	0.4864	0.6162	0.6165
$A_{1,2}$	0.0360	0.0599	0.1106	0.2643	0.4330	0.4925	0.5103
$A_{1,1}$	0.0171	0.0353	0.0996	0.2626	0.4155	0.4862	0.5015
	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	$A_{2,6}$	$A_{2,7}$

6.4.5 Results and Discussion with Simulated Example 4.2

The DOM scores of the resulting FIS models are measured using Equation 3.5 (see Chapter 3 Section 3.4), in which $N_1 = N_2 = 20$ is used. The results for the system identification methods using trapezoidal fuzzy set are depicted in Table 6.17. As an example, as depicted in Table 6.17, for $k_1 = k_2 = 3$, *complete* and *monotone* fuzzy If-Then rules are obtained using system identification method for Stage (B). Therefore, a perfect DOM score of 1.0000 is obtained for Stages (B) and (E), while the SSE values for the training and validation sets are 0.2755 and 0.3131 for Stage B, and 0.2755 and 0.3125 for Stage E, respectively. In short, better SSE values are obtained for Stages (E).

For $k_1 = k_2 = 5$, the DOM values are 0.9159 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.0169 for Stage (B), 0.0522 for Stage (C), and 0.0211 for Stage (E), respectively. However, the SSE values of the validation data set are 1.0462 for Stage (B), 0.1336 for Stage (C), and 0.0993 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 7$, the DOM values are 0.8205 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.0081 for Stage (B), 0.0146 for Stage (C), and 0.0108 for Stage (E). However, the SSE values of the validation data set are 1.4749 for Stage (B), 0.2231 for Stage (C), and 0.1455 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.17: Experimental result using trapezoidal fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	1.0000	0.2755	0.3131	-	-	-	1.0000	0.2755	0.3125
2	5	5	0.9159	0.0169	1.0462	1.0000	0.0522	0.1336	1.0000	0.0211	0.0993
3	7	7	0.8205	0.0081	1.4749	1.0000	0.0146	0.2231	1.0000	0.0108	0.1455

Similar results are obtained using the system identification method for triangular fuzzy set. As an example, as depicted in Table 6.18, for $k_1 = k_2 = 3$, *complete* and *monotone* fuzzy If-Then rules are obtained using system identification method for Stage (B). Therefore, a perfect DOM score of 1.0000 is obtained for Stages (B) and (E), while the SSE values for the training and validation sets are 0.1749 and 0.2417 for Stage B, and 0.1749 and 0.2411 for Stage E, respectively. In short, better SSE values are obtained for Stage (E).

For $k_1 = k_2 = 5$, the DOM values are 0.9659 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.0031 for Stage (B), 0.0032 for Stage (C), and 0.0032 for Stage (E), respectively. However, the SSE values of the validation data set are 0.0138 for Stage (B), 0.0122 for Stage (C), and 0.0124 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 7$, the DOM values are 0.9091 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.0002 for Stage (B), 0.0011 for Stage (C), and 0.0005 for Stage (E). However, the SSE values of the validation data set are 0.1004 for Stage (B), 0.0699 for Stage (C), and 0.0130 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.18: Experimental result using triangular fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	1.0000	0.1749	0.2417	-	-	-	1.0000	0.1749	0.2411
2	5	5	0.9659	0.0031	0.0138	1.0000	0.0032	0.0122	1.0000	0.0032	0.0124
3	7	7	0.9091	0.0002	0.1004	1.0000	0.0011	0.0699	1.0000	0.0005	0.0130

6.5 Results and Discussions

6.5.1 Benchmark Example 1: Thermal Comfort Heat Index Prediction

The DOM scores of the resulting FIS models are measured using Equation 3.5 (see Chapter 3 Section 3.4), in which $N_1 = N_2 = 26$ is used. The experimental results using the system identification method for trapezoidal fuzzy set are summarized in Table 6.19. As an example, for $k_1 = k_2 = 3$, the DOM values are 0.9341 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 21.2051 for Stage (B), 21.6489 for Stage (C), and 21.5274 for Stage (E), respectively. However, the SSE values of the validation data set are 29.9441 for Stage (B), 27.5157 for Stage (C), and 26.3610 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 5$, the DOM values are 0.9396 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 8.5654 for Stage (B), 9.0051 for Stage (C), and 8.7031 for Stage (E), respectively. However, the SSE values of the validation data set are 23.1147 for Stage (B), 16.4857 for Stage (C), and 14.3323 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 7$, the DOM values are 0.9299 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 4.6882 for Stage (B), 7.4363 for Stage (C), and 11.3536 for Stage (E). However, the SSE values of the validation data set are 132.3463 for Stage (B), 30.3377 for Stage (C), and 15.9572 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.19: Experimental result using trapezoidal fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	0.9341	21.2051	29.9441	1.0000	21.6489	27.5157	1.0000	21.5274	26.3610
2	5	5	0.9396	8.5654	23.1147	1.0000	9.0051	16.4857	1.0000	8.7031	14.3323
3	7	7	0.9299	4.6882	132.3463	1.0000	7.4363	30.3377	1.0000	11.3536	15.9572

Similar results are obtained using the system identification method for triangular fuzzy set. As an example, as depicted in Table 6.20, for $k_1 = k_2 = 3$, *complete* and *monotone* fuzzy If-Then rules are obtained using system identification method for Stage (B). Therefore, a perfect DOM score of 1.0000 is obtained for Stages (B) and (E), while the SSE values for the training and validation sets are 0.0108 and 0.0592 for Stage B, and 0.0108 and 0.0582 for Stage E, respectively. In short, better SSE values are obtained for Stage (E).

For $k_1 = k_2 = 5$, the DOM values are 0.9368 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 8.3726e-04 for Stage (B), 0.7567 for Stage (C), and 0.0025 for Stage (E), respectively. However, the SSE values of the validation data set are 15.4794 for Stage (B), 7.9860 for Stage (C), and 0.2312 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 7$, the DOM values are 0.9176 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 1.7651e-04 for Stage (B), 1.0607 for Stage (C), and 0.3362 for Stage (E). However, the SSE values of the validation data set are 32.2724 for Stage (B), 12.5095 for Stage (C), and 4.6081 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.20: Experimental result using triangular fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	1.0000	0.0108	0.0592	-	-	-	1.0000	0.0108	0.0582
2	5	5	0.9368	8.3726e-04	15.4794	1.0000	0.7567	7.9860	1.0000	0.0025	0.2312
3	7	7	0.9176	1.7651e-04	32.2724	1.0000	1.0607	12.5095	1.0000	0.3362	4.6081

The experimental results using the system identification method for 30% noise using trapezoidal fuzzy set are summarized in Table 6.21. As an example, for $k_1 = k_2 = 3$, the DOM values are 0.8764 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 56.7649 for Stage (B), 57.5726 for Stage (C), and 57.3712 for Stage (E), respectively. However, the SSE values of the validation data set are 94.7377 for Stage (B), 88.7786 for Stage (C), and 88.8170 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 5$, the DOM values are 0.8723 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 46.4784 for Stage (B), 49.6872 for Stage (C), and 77.6712 for Stage (E), respectively. However, the SSE values of the validation data set are 122.4524 for Stage (B), 88.8669 for Stage (C), and 83.3165 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 7$, the DOM values are 0.7692 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 37.7215 for Stage (B), 43.1416 for Stage (C), and 52.2958 for Stage (E). However, the SSE values of the validation data set are 249.6570 for Stage (B), 85.3077 for Stage (C), and 64.7705 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.21: Experimental result using trapezoidal fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	0.8764	56.7649	94.7377	1.0000	57.5726	88.7786	1.0000	57.3712	88.8170
2	5	5	0.8723	46.4784	122.4524	1.0000	49.6872	88.8669	1.0000	77.6712	83.3165
3	7	7	0.7692	37.7215	249.6570	1.0000	43.1416	85.3077	1.0000	52.2958	64.7705

Similar results are obtained using the system identification method for 30% noise using triangular fuzzy set. As an example, as depicted in Table 6.22, for $k_1 = k_2 = 3$, the DOM values are 0.9643 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 41.0661 for Stage (B), 41.0812 for Stage (C), and 41.0777 for Stage (E), respectively. However, the SSE values of the validation data set are 55.3183 for Stage (B), 55.3943 for Stage (C), and 55.4122 for Stage (E), respectively.

For $k_1 = k_2 = 5$, the DOM values are 0.8242 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 37.1895 for Stage (B), 41.6855 for Stage (C), and 40.6232 for Stage (E), respectively. However, the SSE values of the validation data set are 119.0556 for Stage (B), 91.6315 for Stage (C), and 57.9738 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 7$, the DOM values are 0.6772 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 34.8787 for Stage (B), 40.5302 for Stage (C), and 41.6019 for Stage (E). However, the SSE values of the validation data set are 109.0262 for Stage (B), 78.3318 for Stage (C), and 82.8536 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.22: Experimental result using triangular fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	0.9643	41.0661	55.3183	1.0000	41.0812	55.3943	1.0000	41.0777	55.4122
2	5	5	0.8242	37.1895	119.0556	1.0000	41.6855	91.6315	1.0000	40.6232	57.9738
3	7	7	0.6772	34.8787	109.0262	1.0000	40.5302	78.3318	1.0000	41.6019	82.8536

The experimental results using the system identification method for 50% noise using trapezoidal fuzzy set are summarized in Table 6.23. As an example, for $k_1 = k_2 = 3$, the DOM values are 0.8599 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 182.2290 for Stage (B), 195.5022 for Stage (C), and 380.0315 for Stage (E), respectively. However, the SSE values of the validation data set are 309.6749 for Stage (B), 264.6312 for Stage (C), and 427.1699 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 5$, the DOM values are 0.7239 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 148.1431 for Stage (B), 208.1125 for Stage (C), and 448.9222 for Stage (E), respectively. However, the SSE values of the validation data set are 744.2674 for Stage (B), 351.0824 for Stage (C), and 444.2816 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 7$, the DOM values are 0.7404 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 133.4195 for Stage (B), 170.8527 for Stage (C), and 183.3136 for Stage (E). However, the SSE values of the validation data set are 1074.3615 for Stage (B), 285.7877 for Stage (C), and 239.7051 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.23: Experimental result using trapezoidal fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	0.8599	182.2290	309.6749	1.0000	195.5022	264.6312	1.0000	380.0315	427.1699
2	5	5	0.7239	148.1431	744.2674	1.0000	208.1125	351.0824	1.0000	448.9222	444.2816
3	7	7	0.7404	133.4195	1074.3615	1.0000	170.8527	285.7877	1.0000	183.3136	239.7051

Similar results are obtained using the system identification method for 50% noise using triangular fuzzy set. As an example, as depicted in Table 6.24, for $k_1 = k_2 = 3$, the DOM values are 0.8036 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 150.2561 for Stage (B), 156.0252 for Stage (C), and 160.9748 for Stage (E), respectively. However, the SSE values of the validation data set are 235.9288 for Stage (B), 220.8258 for Stage (C), and 234.3275 for Stage (E), respectively.

For $k_1 = k_2 = 5$, the DOM values are 0.6511 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 137.5264 for Stage (B), 154.9843 for Stage (C), and 160.5751 for Stage (E), respectively. However, the SSE values of the validation data set are 334.3743 for Stage (B), 250.1303 for Stage (C), and 215.3125 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 7$, the DOM values are 0.5577 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 131.9123 for Stage (B), 157.5366 for Stage (C), and 152.4746 for Stage (E). However, the SSE values of the validation data set are 316.3058 for Stage (B), 217.8737 for Stage (C), and 236.6066 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.24: Experimental result using triangular fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	0.8036	150.2561	235.9288	1.0000	156.0252	220.8258	1.0000	160.9748	234.3275
2	5	5	0.6511	137.5264	334.3743	1.0000	154.9843	250.1303	1.0000	160.5751	215.3125
3	7	7	0.5577	131.9123	316.3058	1.0000	157.5366	217.8737	1.0000	152.4746	236.6066

6.5.2 Benchmark Example 2: Auto-Miles per Gallon (Auto-MPG)

The DOM scores of the resulting FIS models are measured using Equation 3.5 (see Chapter 3 Section 3.4), in which $N_1 = N_2 = 93$ is used. The experimental results using the system identification method for trapezoidal fuzzy set are summarized in Table 6.25. As an example, for $k_1 = k_2 = 3$, the DOM values are 0.7614 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 1.9914e+03 for Stage (B), 6.5451e+03 for Stage (C), and 2.0352e+03 for Stage (E), respectively. However, the SSE values of the validation data set are 1.1519e+04 for Stage (B), 1.0155e+04 for Stage (C), and 1.0515e+04 for Stage (E), respectively. In short, better SSE values are obtained for Stages (C) and (E).

For $k_1 = k_2 = 5$, the DOM values are 0.6249 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 1.6585e+03 for Stage (B), 2.4703e+03 for Stage (C), and 2.2184e+03 for Stage (E), respectively. However, the SSE values of the validation data set are 1.0017e+04 for Stage (B), 7.8835e+03 for Stage (C), and 1.0719e+04 for Stage (E), respectively.

For $k_1 = k_2 = 7$, the DOM values are 0.6527 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 1.6010e+03 for Stage (B), 8.9358e+03 for Stage (C), and 2.3610e+03 for Stage (E). However, the SSE values of the validation data set are 1.0913e+04 for Stage (B), 7.1198e+03 for Stage (C), and 7.2875e+03 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.25: Experimental result using trapezoidal fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	0.7614	1.9914e+03	1.1519e+04	1.0000	6.5451e+03	1.0155e+04	1.0000	2.0352e+03	1.0515e+04
2	5	5	0.6249	1.6585e+03	1.0017e+04	1.0000	2.4703e+03	7.8835e+03	1.0000	2.2184e+03	1.0719e+04
3	7	7	0.6527	1.6010e+03	1.0913e+04	1.0000	8.9358e+03	7.1198e+03	1.0000	2.3610e+03	7.2875e+03

Similar results are obtained using the system identification method for triangular fuzzy set. As an example, as depicted in Table 6.26, for $k_1 = k_2 = 3$, the DOM values are 0.7338 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 1.8615e+03 for Stage (B), 1.8915e+03 for Stage (C), and 1.8861e+03 for Stage (E), respectively. However, the SSE values of the validation data set are 1.0280e+04 for Stage (B), 8.5841e+03 for Stage (C), and 8.9210e+03 for Stage (E), respectively.

For $k_1 = k_2 = 5$, the DOM values are 0.3935 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 1.7617e+03 for Stage (B), 2.3578e+03 for Stage (C), and 3.4497e+04 for Stage (E), respectively. However, the SSE values of the validation data set are 9.9199e+03 for Stage (B), 6.5674e+03 for Stage (C), and 1.1191e+04 for Stage (E), respectively.

For $k_1 = k_2 = 7$, the DOM values are 0.3252 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 1.5633e+03 for Stage (B), 5.2255e+03 for Stage (C), and 1.9013e+03 for Stage (E). However, the SSE values of the validation data set are 9.6913e+03 for Stage (B), 3.1536e+03 for Stage (C), and 6.5103e+03 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.26: Experimental result using triangular fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	0.7338	1.8615e+003	1.0280e+004	1.0000	1.8915e+003	8.5841e+003	1.0000	1.8861e+003	8.9210e+003
2	5	5	0.3935	1.7617e+003	9.9199e+003	1.0000	2.3578e+003	6.5674e+003	1.0000	3.4497e+004	1.1191e+004
3	7	7	0.3252	1.5633e+003	9.6913e+003	1.0000	5.2255e+003	3.1536e+003	1.0000	1.9013e+003	6.5103e+003

6.5.3 Benchmark Example 3: Mercury-in-Fish

The DOM scores of the resulting FIS models are measured using Equation 3.5 (see Chapter 3 Section 3.4), in which $N_1 = N_2 = 12$ is used. The experimental results using the system identification method for trapezoidal fuzzy set are summarized in Table 6.27. As an example, for $k_1 = k_2 = 3$, the DOM values are 0.6607 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.6091 for Stage (B), 0.7038 for Stage (C), and 0.8041 for Stage (E), respectively. However, the SSE values of the validation data set are 0.3131 for Stage (B), 0.3381 for Stage (C), and 0.3529 for Stage (E), respectively.

For $k_1 = k_2 = 5$, the DOM values are 0.4583 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.4344 for Stage (B), 1.1182 for Stage (C), and 1.0111 for Stage (E), respectively. However, the SSE values of the validation data set are 0.3803 for Stage (B), 0.7861 for Stage (C), and 0.4080 for Stage (E), respectively.

For $k_1 = k_2 = 7$, the DOM values are 0.3869 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.4449 for Stage (B), 0.6467 for Stage (C), and 0.8940 for Stage (E). However, the SSE values of the validation data set are 0.3395 for Stage (B), 0.3824 for Stage (C), and 0.4486 for Stage (E). Again, better SSE values are obtained for Stages (C) and (E).

Table 6.27: Experimental result using trapezoidal fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	0.6607	0.6091	0.3131	1.0000	0.7038	0.3381	1.0000	0.8041	0.3529
2	5	5	0.4583	0.4344	0.3803	1.0000	1.1182	0.7861	1.0000	1.0111	0.4080
3	7	7	0.3869	0.4449	0.3395	1.0000	0.6467	0.3824	1.0000	0.8940	0.4486

Similar results are obtained using the system identification method for triangular fuzzy set. As an example, as depicted in Table 6.28, for $k_1 = k_2 = 3$, the DOM values are 0.3214 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.5356 for Stage (B), 0.9515 for Stage (C), and 0.6483 for Stage (E), respectively. However, the SSE values of the validation data set are 0.3078 for Stage (B), 0.5054 for Stage (C), and 0.3591 for Stage (E), respectively.

For $k_1 = k_2 = 5$, the DOM values are 0.3690 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.4474 for Stage (B), 1.4347 for Stage (C), and 0.6184 for Stage (E), respectively. However, the SSE values of the validation data set are 0.3077 for Stage (B), 0.8099 for Stage (C), and 0.3456 for Stage (E), respectively.

For $k_1 = k_2 = 7$, the DOM values are 0.4167 for Stage (B) and 1.0000 for Stages (C) and (E), respectively. The SSE values of the training data set are 0.4315 for Stage (B), 0.9758 for Stage (C), and 0.6877 for Stage (E). However, the SSE values of the validation data set are 0.3353 for Stage (B), 0.5425 for Stage (C), and 0.3490 for Stage (E).

Table 6.28: Experimental result using triangular fuzzy set

	k_1	k_2	Without Relabeling			With Relabeling					
						Solution 1			Solution 2		
			DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data	DOM	SSE with training data	SSE with validation data
1	3	3	0.3214	0.5356	0.3078	1.0000	0.9515	0.5054	1.0000	0.6483	0.3591
2	5	5	0.3690	0.4474	0.3077	1.0000	1.4347	0.8099	1.0000	0.6184	0.3456
3	7	7	0.4167	0.4315	0.3353	1.0000	0.9758	0.5425	1.0000	0.6877	0.3490

6.5.4 Remarks

In short, these experimental results indicate that while the training and validation data sets are *non-monotone* (noisy), by including Stages (C) or Stage (E), a perfect DOM score can be obtained. Interestingly, some SSE values for the validation data set are better with the addition of Stages (C) or Stage (E), as compared with those from Stage (B). The outcome ascertains that the monotonicity property and Proposition 1 are important additional information for developing *monotone* FIS models.

6.6 Summary

In this chapter, an analysis of the system identification method for single and multi-attribute *monotone* data sets is presented. A system identification-based framework to develop *monotone* FIS TSK models with system identification method is proposed. A series of benchmark problem are conducted to demonstrate the usefulness of our proposed approach.

In the next chapter, a conclusion remarks and future works are discussed thoroughly.

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

The focus of this thesis is on *data-driven monotone FIS models* where the *sufficient conditions as a set of governing equation for designing and developing monotonicity-preserving FIS models* are established. A *monotone* test is used to evaluate the monotonicity property of an FIS model. An analysis for the data-driven FIS models, i.e., *ad hoc* and system identification methods are conducted. Our study shows that fuzzy If-Then rules generated using both *ad hoc* and system identification methods are always *monotone*, when single-attribute *monotone* data are employed. However, this is not the case for multi-attribute monotone data. This is an important finding, as it indicates that even with a "clean" (i.e., *monotone*) data set, the generated fuzzy If-Then rules may need to be pre-processed (re-labeled), in order to satisfy the monotonicity property.

Thus, a monotonicity data-driven preserving FIS framework model for both *ad hoc* and system identification methods are developed. As such for *ad hoc* method, a framework for practical modeling of *monotone* TSK FIS models with *ad hoc* methods, together with a *monotone* fuzzy rules relabeling (MFRR) procedure, is formulated. On the other hand, for system identification method, two formulations, i.e., (1) based on MFRR; and (2) based on a constrained derivative-based optimization method, are devised.

A real-world case study has been conducted to vindicate the usefulness of the proposed framework, and positive results have been obtained. The results demonstrate that given a *monotone* data set, both proposed methods lead to better accuracy (i.e., a lower sum square error) and DOM of the resulting TSK FIS model.

7.2 Limitations

In this project, only convex and normal trapezoidal and triangular fuzzy sets are used as the fuzzy membership functions, with a strong fuzzy partition strategy is investigated. Besides that, only a fixed fuzzy partition is considered in this project.

7.3 Future Recommendation

Various constrained mathematical programming methods, e.g., Sequential Quadratic Programming (Boggs & Tolle, 1995) and meta-heuristic methods, e.g., Genetic Algorithm (Boggs & Tolle, 1995), Particle Swarm Optimization (Kennedy & Eberhart, 1995), Harmony Search (Geem et al., 2001), and etc can be used to solve the constrained non-linear problem. As future work, a data-driven *monotone* TSK FIS paradigm with adaptive fuzzy partition will be extended along with the multi-objective optimization (Geem et al., 2001), where two objective functions will be considered, i.e., minimizing the error function (best-fit of a data set) and minimizing the number of fuzzy rules. In addition, learning locally *monotone* or near *monotone* TSK FIS models will be studied. The aim is to formulate a robust system identification framework for modelling useful and usable TSK FIS models that guarantee the monotonicity property for practical applications.

REFERENCES

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. Retrieved from http://www.ics.uci.edu/~mlearn/MLRepository.html

Baranyi, P., Kóczy, L. T., & Gedeon, T. D. (2004). A generalized concept for fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, *12*(6), 820–837.

Boggs, P. T., & Tolle, J. W. (1995). Sequential Quadratic Programming. *Acta Numerica*, *4*, 1-51.

Box, G. E. P., & Tiao, G. C. (2011). *Bayesian inference in statistical analysis*. New York: John Wiley & Sons.

Broekhoven, E. V., & Baets, B. D. (2009). Only smooth rule bases can generate monotone Mamdani–Assilian models under center-of-gravity defuzzification. *IEEE Transactions on Fuzzy Systems*, *17*(5), 1157–1174.

Casillas, J., Cordón, O., & Herrera, F. (2002). COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *32*(4), 526–537.

Casillas, J., Cordón, O., Herrera, F., & Magdalena, L. (2003). *Accuracy improvements in linguistic fuzzy modeling*. Berlin: Springer.

Chang, Y. C., Chen, S. M., & Liau, C. J. (2008). Fuzzy interpolative reasoning for sparse fuzzy-rule-based systems based on the areas of fuzzy sets. *IEEE Transactions on Fuzzy Systems*, *16*(5), 1285–1301.

Chong;, E. K. P., & Zak, S. H. (2004). *An introduction to optimization* (2nd Edition). New York: Wiley.

Cooper, F. G. (1990). The computational complexity of probabilistic inference using bayesian

belief networks. *Artificial Intelligence*, *43*(2–3), 393–405.

Cordón, O., Herrera, F., & Peregrín, A. (1997). Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy Sets and Systems, 86*(1), 5–41.

Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, *60*(1), 141–153.

Daniels, H., & Velikova, M. (2003). Derivation of monotone decision models from non-monotone data, *Tilburg University, Tilburg, The Netherlands, Center for Economic Research, 2003*.

Daniels, H., & Velikova, M. (2010). Monotone and partially monotone neural networks. *IEEE Transactions on Neural Networks*, *21*(6), 906–917.

Davidon, W. C. (1991). Variable metric method for minimization. *SIAM Journal on Optimization*, *1*(1), 1–17.

Dempster, A. P. (1968). A generalization of Bayesian inference. *Journal of the Royal Statistical Society. Series B*, *30*(2), 205–247.

Du, H., & Zhang, N. (2008). Application of evolving Takagi–Sugeno fuzzy model to nonlinear system identification. *Applied Soft Computing*, *8*(1), 676–686.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-seperated clusters. *Journal of Cybernetics*, *3*(3), 32–57.

Emami, M. R., Turksen, I. B., & Goldenberg, A. A. (1999). A unified parameterized formulation of reasoning in fuzzy modeling and control. *Fuzzy Sets and Systems*, *108*(1), 59–81.

Fanger, P. O. (1970). *Thermal comfort: analysis and applications in environmental engineering*. New York : McGraw-Hill.

Feng, G. (2006). A survey on analysis and design of model-based fuzzy control systems. *IEEE Transactions on Fuzzy Systems*, *14*(5), 676–697.

Fletcher, R., & Powell, M. J. D. (1963). A rapidly convergent descent method for minimization. *The Computer Journal*, *6*(2), 163–168.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, *76*(2), 60–68.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Boston, MA, USA : Addison-Welsey.

Goldreich, O., Goldwasser, S., Lehman, E., Ron, D., & Samorodnitsky, A. (2000). Testing monotonicity. *Combinatorica*, *20*(3), 301–337.

Hardy, G., Littlewood, J., & Polya, G. (1934). *Inequalities*. London: Cambridge University Press.

Harrison, J. (2009). *Handbook of practical logic and automated reasoning*. New York: Cambridge University Press.

Hinton, G. E., & Sejnowski, T. (1983). Optimal perceptual inference. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 448–453).

Hoffmann, F. (2001). Evolutionary algorithms for fuzzy control system design. *Proceedings of the IEEE*, *89*(9), 1318–1333.

Huang, Z., & Shen, Q. (2006). Fuzzy interpolative reasoning via scale and move transformations. *IEEE Transactions on Fuzzy Systems*, *14*(2), 340–359.

Husek, P. (2016). On monotonicity of Takagi-Sugeno fuzzy systems with ellipsoidal regions. *IEEE Transactions on Fuzzy System*, *24*(6), 1673–1678.

Ichihashi, H., & Watanabe, T. (1990). Learning control system by a simplified fuzzy reasoning model. In *Proceeding of IPMU (1990)* (pp. 417–419).

International Council for the Exploration of the Sea (ICES) database. (n.d.). Statistical analysis of the ICES co-operative monitoring programme data on contaminants in fish muscle tissue (1975–1985) for determination of temporal trends. *ICES Cooperative Research Report No.162*. Retrieved from http://www.ices.dk

Ishibuchi, H., Nozaki, K., Tanaka, H., Hosaka, Y., & Matsuda, M. (1993). Empirical study on learning in fuzzy systems. In *2nd IEEE International Conference on Fuzzy Systems, 1993*.

Ishibuchi, H., Nozaki, K., Tanaka, H., Hosaka, Y., & Matsuda, M. (1994). Empirical study on learning in fuzzy systems by rice taste analysis. *Fuzzy Sets and Systems*, *64*(2), 129–144.

Ishibuchi, H., Nozaki, K., Yamamoto, N., & Tanaka, H. (1995). Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, *3*(3), 260–270.

Ishibuchi, H., Yamamoto, T., & Nakashima, T. (2006). An approach to fuzzy default reasoning for function approximation. *Soft Computing*, *10*(9), 850–864.

Jang, J. S. R. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*(3), 665–685.

Jang, J. S. R., & Sun, C. T. (1995). Neuro-fuzzy modeling and control. *The Proceedings of the IEEE*, *83*(3), 378–406.

Jang, J. S. R., Sun, C. T., & Mizutani, E. (1997). *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*. USA: Prentice-Hall.

Jee, T. L., Tay, K. M., & Lim, C. P. (2015). A new two-stage fuzzy inference system-based

approach to prioritize failures in failure mode and effect analysis. *IEEE Transactions on Reliability*, *64*(3), 869–877.

Jenei, S. (2001). Interpolation and extrapolation of fuzzy quantities revisited – an axiomatic approach. *Soft Computing*, *5*(3), 179–193.

Jenei, S., Klement, E. P., & Konzel, R. (2002). Interpolation and extrapolation of fuzzy quantities–The multiple-dimensional case. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, *6*(3), 258–270.

Jin, Y. (2000). Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement. *IEEE Transactions on Fuzzy Systems*, *8*(2), 212–221.

Kahane, H. (1990). *Logic and philosophy : A modern introduction by Howard Kahane* (11th Edition). Australia: Wadsworth.

Kandel, A. (1992). *Fuzzy expert systems*. New York, USA: CRC Press.

Karnik, N. N., Mendel, J. M., & Liang, Q. (1999). Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, *7*(6), 643–658.

Kaufman, P. J. (1998). *Trading Systems and Methods*. New York, USA: Wiley.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks IV* (pp. 1942–1948).

Kerk, Y. W., Tay, K. M., & Lim, C. P. (2017). An analytical interval fuzzy inference system for risk evaluation and prioritization in failure mode and effect analysis. *IEEE Systems Journal*, *11*(3), 1589–1600.

Kneebone, G. T. (2001). *Mathematical logic and the foundation of mathematics: an introductory survey*. New York, USA: Dover Publications.

Ko, Y. K., & Chen, S. M. (2007). Fuzzy interpolative reasoning via cutting and transformations techniques. In *International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems, 2007* (pp. 238–249).

Koczy, L., & Hirota, K. (1993a). Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, *9*(3), 197–225.

Koczy, L., & Hirota, K. (1993b). Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Sciences*, *71*(1–2), 169–201.

Koczy, L., & Hirota, K. (1997). Size reduction by interpolation in fuzzy rule bases. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 27*(1), 14–25.

Kóczy, L. T., & Hirota, K. (1993). Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, *9*(3), 197–225.

Kosko, B. (1990). Fuzzy associative memories. In *Neural network and fuzzy systems*. New York, USA: Prentice-Hall.

Kosko, B. (1994). Fuzzy systems as universal approximators. *IEEE Transactions on Computers*, *43*(11), 1329–1333.

Kouikoglou, V. S., & Phillis, Y. A. (2009). On the monotonicity of hierarchical sum-product fuzzy systems. *Fuzzy Sets System*, *160*(24), 3530–3538.

Kurnaz, S., Cetin, O., & Kaynak, O. (2010). Expert systems with applications adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. *Expert Systems With Applications*, *37*(2), 1229–1234.

Lau, P., & Chan, C. W. (1997). An enhanced weighted fuzzy reasoning algorithm for engineering applications. *Computers & Mathematics with Applications*, *34*(1), 81–87.

Lee, C. C. (1990a). Fuzzy logic in control systems: fuzzy logic controller. I. *IEEE Transactions on Systems, Man, and Cybernetics*, *20*(2), 404–418.

Lee, C. C. (1990b). Fuzzy logic in control systems: fuzzy logic controller. II. *IEEE Transactions on Systems, Man, and Cybernetics*, *20*(2), 419–435.

Li, C., Yi, J., Wang, M., & Zhang, G. (2013). Monotonic type-2 fuzzy neural network and its application to thermal comfort prediction. *Neural Computing and Applications*, *23*(7–8), 1987–1998.

Liang, Q. L., & Mendel, J. (2000). Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy Systems*, *8*(5), 535–550.

Lin, C. T., & Lee, C. S. G. (1995). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent systems*. New York, USA: Prentice-Hall.

Lindskog, P., & Ljung, L. (2000). Ensuring monotonic gain characteristics in estimated models by fuzzy model structures. *Automatica*, *36*(2), 311–317.

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, *7*(1), 1–13.

Markowski, A. S. (2005). The implementation of the SEVESO II legislation in the Polish major hazard industry. *Journal of Loss Prevention in the Process Industries*, *18*(4–6), 360–364.

Markowski, A. S., & Mannan, M. S. (2008). Fuzzy risk matrix. *Journal of Hazardous Materials*, *159*(1), 152–157.

Mendel, J. M. (1995). Fuzzy logic systems for engineering: A tutorial. *Proceedings of the IEEE*, *83*(3), 345–377.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd Edition). New York, USA:

Springer-Verlag.

Nomura, H., Hayashi, I., & Wakami, N. (1992). A learning method of fuzzy inference rules by descent method. In *Proceeding of FUZZ-IEEE 1992* (pp. 203–210).

Nozaki, K., Ishibuchi, H., & Tanaka, H. (1997). A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems*, *86*(3), 251–270.

Oderanti, F. O., & De Wilde, P. (2010). Dynamics of business games with management of fuzzy rules for decision making. *International Journal of Production Economics*, *128*(1), 96–109.

Pang, L. M., Tay, K. M., & Lim, C. P. (2016). Monotone fuzzy rule relabeling for the zero-order TSK fuzzy inference system. *IEEE Transactions on Fuzzy Systems*, *24*(6), 1455–1463.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. New York, USA: Morgan Kaufman.

Rademaker, M., De Baets, B., & De Meyer, H. (2009). Loss optimal monotone relabeling of noisy multi-criteria data sets. *Information Sciences*, *179*(24), 4089–4096.

Raha, S., Pal, N. R., & Ray, K. S. (2002). Similarity-based approximate reasoning: methodology and application. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *32*(4), 541–547.

Restrepo, A., & Bovik, A. C. (1993). Local monotonic regression. *IEEE Transactions on Signal Processing*, *41*(9), 1796–2810.

Russell, S. J., & Norvig, P. (2003). *Artificial intelligence a modern approach* (3rd Edition.). New York, USA: Prentice-Hall.

Sánchez, E., & Zadeh, L. A. (1987). *Approximate reasoning in intelligent systems, decision and control*. United Kingdom: Pergamon Press.

Seki, H., Ishii, H., & Mizumoto, M. (2010). On the monotonicity of fuzzy inference methods related to T-S inference method. *IEEE Transaction on Fuzzy Systems*, *18*(3), 629–634.

Sengur, A. (2008). Wavelet transform and adaptive neuro-fuzzy inference system for color texture classification. *Expert Systems with Applications*, *34*(3), 2120–2128.

Stamatis, D. H. (2003). *Failure mode and effect analysis: FMEA from theory to execution*. New York, USA: ASQ Quality Press.

Stoll, R. R. (1975). *Sets, Logic and Axiomatic Theories (English and Multilingual Edition)* (2nd Edition). San Francisco: W.H.Freeman & Co Ltd.

Sugeno, M., & Kang, G. T. (1988). Structure identification of fuzzy model. *Fuzzy Sets System*, *28*(1), 15–33.

Sugeno, M., & Yasukawa, T. (1993). A fuzzy-logic-based approach to qualitative modelling. *IEEE Transaction on Fuzzy Systems*, *1*(1), 7–31.

Takagi, H., & Hayashi, I. (1991). NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning*, *5*(3), 191–212.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, *15*(1), 116–132.

Tay, K. M., & Lim, C. P. (2008). On the use of fuzzy inference techniques in assessment models: Part II: Industrial applications. *Fuzzy Optimization and Decision Making*, *7*(3), 283–302.

Tay, K. M., & Lim, C. P. (2011a). An evolutionary-based similarity reasoning scheme for monotonic multi-input fuzzy inference systems. In *IEEE -FUZZ (2011)* (pp. 442–447).

Tay, K. M., & Lim, C. P. (2011b). On monotonic sufficient conditions of fuzzy inference systems and their application. *International Journal of Uncertainty, Fuzziness and Knowledge-Based*

Systems, *19*(5), 731–757.

Tay, K. M., & Lim, C. P. (2011c). Optimization of Gaussian fuzzy membership functions and evaluation of the monotonicity property of fuzzy inference systems. In *IEEE-FUZZ (2011)* (pp. 1219–1224).

Tikk, D., & Baranyi, P. (2000). Comprehensive analysis of a new fuzzy rule interpolation method. *IEEE Transactions on Fuzzy Systems*, *8*(3), 281–296.

Tsang, E. C. C., & Yeung, D. S. (1997). Modelling fuzzy production rules with fuzzy expert networks. *Expert Systems with Applications*, *13*(3), 169–178.

Tsukamoto, Y. (1979). An approach to fuzzy reasoning method. In M. M. Gupta, R. K. Ragade, & R. R. Yager Edition, *Advances in Fuzzy Set Theory and Applications* (pp. 137–149).

Turksen, I. B., & Zhao, Z. (1988). An approximate reasoning approach for the implementation. In *IEEE International Conference on Systems, Man, and Cybernetics, 1988.* (pp. 173–176).

Wang, L. X. (1992). Fuzzy systems are universal approximators. In *IEEE International Conference on Fuzzy Systems* (pp. 1163–1170).

Wang, L. X. (2003). The WM method completed: a flexible fuzzy system approach to data mining. *IEEE Transactions on Fuzzy Systems*, *11*(6), 768–782.

Wang, L. X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, *22*(6), 1414–1427.

Won, J. M., Park, S. Y., & Lee, J. S. (2002). Parameter conditions for monotonic Takagi-Sugeno-Kang fuzzy system. *Fuzzy Sets and Systems*, *132*(2), 135–146.

Wong, K. W., Tikk, D., Gedeon, T. D., & Kóczy, L. T. (2005). Fuzzy rule interpolation for multidimensional input spaces with applications: A case study. *IEEE Transactions on Fuzzy*

Systems, *13*(6), 809–819.

Wu, C. J., & Sung, A. H. (1996). A general purpose fuzzy controller for monotone functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *26*(5), 803–808.

Yam, Y., Wong, M. L., & Baranyi, P. (2006). Interpolation with function space representation of membership functions. *IEEE Transactions on Fuzzy Systems*, *14*(3), 398–411.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-3*(1), 28–44.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*, *8*(3), 199–249.

Zhao, H., & Zhu, C. (2000). Monotone fuzzy control method and its control performance. In *IEEE International Conference on Systems, Man, and Cybernetics* (pp. 3740–3745).

APPENDIX

List of Publications

Journal Paper

Teh, C. Y., Kerk, Y. W., Tay, K. M., & Lim C. P. (2018). On Modelling of Data-Driven Monotone Zero-Order TSK Fuzzy Inference Systems using a System Identification Framework. *IEEE Transaction on Fuzzy Systems*. Accepted. (Impact Factor: 8.415)

Teh, C. Y., Tay, K. M., & Lim C. P. (2018). On the Monotonicity Property of the TSK Fuzzy Inference System: The Necessity of the Sufficient Conditions and the Monotonicity Test. *International Journal of Fuzzy Systems*, 20(6), 1915-1924. (Impact Factor: 2.396).

Book Chapter

Teh, C. Y., & Tay, K. M. (2017). Monotone Data Samples Do Not Always Generate Monotone Fuzzy If-Then Rules. *Emerging Trends in Neuro Engineering and Neural Computation*, 255-264.

Conference Papers

Teh, C. Y., Tay, K. M., & Lim, C. P. (2017). Monotone data samples do not always produce monotone fuzzy if-then rules: Learning with ad hoc and system identification methods. *IEEE International Conference on Fuzzy System (FUZZ), Naples, Italy, 2017*.

Teh, C. Y., Tay, K. M., & Lim, C. P. (2013). Interval-based and fuzzy set-based approaches to modeling of fuzzy inference systems with the local monotonicity

property. *IEEE International Conference on Fuzzy System (FUZZ), Hyderabad, India, 2013.*

Tay, K. M., Lim, C. P., Teh, C. Y., & Lau, S. H. (2012). A monotonicity index for the monotone fuzzy modeling problem. *IEEE International Conference on Fuzzy System (FUZZ), Brisbane, QLD, Australia, 2012.*