Handling Uncertainty: From Type-1 to Interval Type-2 Fuzzy Sets and Systems

Jabran Hussain Aladi, BSc. MSc.

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

March 2016

This thesis is dedicated to my parents

Abstract

Fuzzy logic systems (FLSs) are widely accepted for their ability to model and handle uncertainty. Type-2 fuzzy sets (T2 FSs) were introduced as an extension of type-1 fuzzy sets (T1 FSs). They are characterised by membership functions (MFs) that are themselves fuzzy for which the membership degrees are expressed as FSs on [0,1], and have been widely accepted as capable of modelling uncertainty with more detail than T1 FSs. Interval type-2 fuzzy sets (IT2 FSs) are a special type of (general) T2 FSs and currently the most widely used, due to their reduction in computational cost. The study of T2 FLSs is a rapidly growing research area with a wide range of application domains.

Capturing the uncertainty arising from system noise has been a core feature of FLSs for many years. Since the concept of T2 FSs was introduced, a recurring question in research considering the application of T2 FSs is,'*How much* uncertainty in a given context warrants the use of T2 FSs and systems over their T1 counterparts?' In other words, while a main issue in the application of FLSs is the estimation of parameters such as the type of fuzzy sets (FSs) and their parameters, as well as the number of rules, an even more fundamental question is whether T1 or T2 FSs should be used. More specifically, 'How should T2 FSs be shaped in order for them to capture the uncertainties in a given application?' Although there is experimental evidence showing improvements in terms of the uncertainty handling of interval type-2 fuzzy logic systems (IT2 FLSs) over their T1 counterparts, no systematic way of determining the potential advantages of employing T2 FLSs over T1 has yet been developed.

In an effort to relate the size of the footprint of uncertainty (FOU) of employed IT2 FSs to uncertainty levels and vice versa in a given application, this thesis shows

the relationship between the size of the FOU of IT2 FSs and the uncertainty levels in a given application and explains how this knowledge can be exploited to inform the design of FLSs. To provide insight into this challenging aim, a detailed investigation of the ability of both T1 and IT2 FLSs to model different levels of uncertainty/noise is conducted. Design methodologies that systematically vary (blur) the size of the FOU of the IT2 FSs are introduced, enabling the comparison of FLSs that are equivalent in all but the size of the FOUs of the employed FSs.

We describe an application-driven investigation into the relationship between the FOU size of the FSs and the level of uncertainty in applications by using time series prediction (TSP) as a well-defined and well-controlled sample application. Thus, TSP is used as a platform to comprehensively compare different FLSs with various FOUs. Through contrasting the performance of these resulting FLSs in the face of inputs with varying uncertainty levels in a rich set of TSP experiments, a distinct pattern of performance arising from the different levels-of-uncertainty and FOU-size combinations is explored and captured, showing a direct relationship between FOU size and uncertainty levels. For example, as the noise level increases, the FOU size that gives the best performance increases. Based on this, we provide guidelines for the selection of appropriate FOU sizes for given levels of uncertainty in a given application and propose an approach to quantifying the commonly used linguistic labels, 'low', 'medium' and 'high' through FS models.

Finally, going beyond the question of selecting the most appropriate FOU at design time, we conduct some initial work on the appropriate adjustment of FOUs at run time, i.e., when uncertainty levels vary. Specifically, we explore the application of optimisation methods to refine FOU sizes in IT2 FSs.

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful

Alhamdulillah, all praises to Allah Almighty for His blessing and granting me the chance, health, strength, ability and time to complete this thesis.

First of all, I would like to express my gratitude to my supervisors, Christian Wagner and Jon Garibaldi, for their guidance, continuous and constant support, encouragement, motivation and constructive comments and suggestion during the entire journey of conducting this research.

I would like to thank my office colleagues, Simon Miller, Ian Dent, Alexandros Ladas and Robert Miles for being interesting, helpful and great companions.

My thanks also goes to Jerry Mendel, for his constructive comments on our paper presented in WCCI2014.

I would like to thank my examiners, Professor Qiang Shen and Professor, Robert John for their fair assessment of the thesis and their constructive comments for improvement.

I would like to thank the Ministry of Defence in Saudi Arabia for sponsoring me and offering me this opportunity to study for a PhD. I also thank the Saudi Arabia Cultural Bureau in the United Kingdom, for their support and valuable advice during my study. Both sponsors were helpful and providing a great service whenever I require one.

I thank all of my friends, in Saudi Arabia, the UK and elsewhere around the world for their valuable friendship and for the support throughout the years that made my life in the UK much easier. My special thanks goes to my brother-in-law, Hussain Alqahtani for his endless support and encouragements during my stay in UK. Also, my special thanks to my friends, Saeed Alqahtani, Mazin Bukhari, Fawaz Alanzi and Sultan Alahmri for their special care, encouragements and support during my study.

I wish to thank all of my family in Saudi Arabia for their support and encouragement, and especially my mother. Although several thousand miles away, their unconditional love made my study period here much more bearable.

I would like to show deep gratefulness to my wife, Hind, without whom this effort would have been worth nothing. For her sacrifice, patience, supports, unlimited love, and accompany throughout the PhD journey. My endless love goes to my children, Ghadi, AbdulSalam, AbdulWahab, Ghala and Mohammed, for their patience and filling my life with happiness. The nights away from the family while attending conferences and the busy days while conducting research and writing this thesis were truly difficult.

Finally, my thanks go to all the people who have supported me in any respect during the completion of the thesis.

Abbreviations and their Definitions

Abbreviation	Definition
CB	Controlled Blurring
COG	Centre of Gravity
COS	Centre of Sets
dB	Decibel
DOU	Domain Of Uncertainty
FLC	Fuzzy Logic Control
FLS	Fuzzy Logic System
FOU	Footprint Of Uncertainty
FS	Fuzzy Set
GT2 FS	General Type-2 Fuzzy Set
IK	Ikeda (time series)
IT2 FLS	Interval Type-2 Fuzzy Logic System
IT2 FS	Interval Type-2 Fuzzy Set
KM	Karnik-Mendel
LMF	Lowe Membership Function
LZ	Lorenz (time series)
MF	Membership Function
MG	Mackey-Glass (time series)
MISO	Multiple Inputs Single Output
NF	Noise Free
NFTrain	Noise Free Training

Abbreviation	Definition
NSFLS	Non-Singleton Fuzzy Logic System
NTrain	Noisy Training
O-CB	Optimised Controlled Blurring
RMSE	Root Mean Squared Error
SA	Simulated Annealing
SB	Standard Blurring
SFLS	Singleton Fuzzy Logic System
SNR	Signal-to-Noise Ratio
SP-CB	Shape-Preserving Controlled Blurring
T1 FLS	Type-1 Fuzzy Logic System
T1 FS	Type-1 Fuzzy Set
T2 FLS	Type-2 Fuzzy Logic System
T2 FS	Type-2 Fuzzy Set
TSP	Time Series Prediction
UMF	Upper Membership Function
WM	Wang-Mendel

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Uncertainty exists in almost all real life situations. In general, uncertainty is an attribute of information and emerges whenever a situation is imprecise, noisy, vague or incomplete. Therefore, it is crucial for intelligent systems working in complex and dynamic environments to have the ability to handle uncertain information and to reason with incomplete knowledge. Frameworks such as probability theory and fuzzy logic sets and systems are established concepts and techniques that are used for modelling uncertainty in such environments. This thesis is concerned with modelling uncertainties with fuzzy sets (FSs) and fuzzy logic systems (FLSs). The choice of FLSs is based on their attractive capability to model variables in human-like terms that are seen through the introduction of the linguistic variables. Additionally, it has been shown that T2 FLSs as an extensions of T1 FLSs have the ability to provide a more fine-grained model of uncertainty.

Since type-1 fuzzy sets (T1 FSs) were introduced by Zadeh in 1965 [1], FLSs have been accepted as a methodology for building systems that can deliver excellent performance in the face of uncertainty and imprecision [2–5]. FLSs have been successfully implemented in many real world applications, including modelling and control [6],[7], the forecasting of time series [8–10] and data mining [11],[12].

Type-1 fuzzy logic systems (T1 FLSs) that operate on T1 FSs are the most well-known and widely used type of FLS. However, T1 FLSs, when compared to type-2 fuzzy logic systems (T2 FLSs), only handle a limited level of uncertainty (when using the same number of FSs). Real-world applications are often faced with multiple sources of high and/or varying levels of uncertainty [13]. In 1975, Zadeh [14] recognized this potential limitation and introduced the concept of type-2 fuzzy sets (T2 FSs) as an extension to T1 FSs. T2 FSs, as more complex models, are considered more suited to modelling uncertainty. Their additional complexity arises from the inclusion of a footprint of uncertainty (FOU) and a third dimension, offering extra degree of freedom to T2 FSs in comparison to T1 FSs [13],[15] (i.e., T2 FSs are described by MFs that are characterized by more parameters than are MFs for T1 FSs [16]). While the latter effectively means that an optimally configured General T2 (GT2) FLS will provide at least the same performance as an equivalent (in terms of number of FSs and rules) optimally configured IT2 FLS which in turn will provide at least the same performance as an equivalent optimally configured T1 FLS [17], it is the same complexity which makes T2 FLSs computationally demanding in comparison to those that employ T1 FSs.

The computational and design complexity of general T2 FLSs has been widely investigated recently [5],[18–27]. Generally, the computational complexity of general

Table 1.1 Sample articles citing the presence of 'large amounts of uncertainty' in their applications as the reason for employing T2 FLS.

Author(s)	Ref. no.	Quote
Lin et al.	[32]	'Due to the rule uncertainties and the training data corrupted by noise, the circumstances are **too uncertain** to determine exact membership grades. A new direct adaptive interval type-2 fuzzy controller is developed to handle such uncertainties for a class of multi-variable non-linear dynamical systems involving external disturbances'
Wagner and Hagras	[33]	'The type-2 Fuzzy Logic Controller (FLC) has started to emerge as a promising control mechanism for autonomous mobile robots navigating in real world environments. This is because such robots need control mechanisms such as type-2 FLCs which can **handle the large amounts of uncertainty** present in real world environments'.
Baklouti and Alimi	[34]	'Motion planning of mobile robots in unknown and dynamic environments is faced with a **large amount of uncertainty**'.

T2 FLSs has favoured the application of simplified interval type-2 fuzzy logic systems (IT2 FLSs) [28–31] which today are the most commonly used kind of T2 FLS. IT2 FLSs employ IT2 FSs, which are a special case of a GT2 FSs where all the secondary membership grades are equal to one and therefore only differentiate themselves from T1 FSs by their FOU.

Many researchers favour IT2 FLSs over T1 FLSs because of their potential to model and handle the effects of uncertainty [35–37] while maintaining the number of FSs and rules constant. To illustrate, Table 1.1 provides a sample of some articles which directly cite the presence of ‘large amounts of uncertainty’ in their given applications as the reason for employing T2 FSs.

In terms of the actual application of T2 FLSs, a recent review of industrial applications of T2 FLSs and FSs by Dereli et al. [38] showed that there were different reasons that the studies chose T2 FLSs for their applications. The authors summarise these reasons in three groups. (1) ‘To handle more uncertainty’, (2) ‘To investigate the effectiveness of type-2 FSs and to compare with type-1 FSs’ and (3) ‘The non-existence of an approach previously employing them for a certain problem type’ [38]. It is important to note that the authors of many of the papers reviewed in [38] that considered T2 FSs in their studies aimed to design systems that would perform well in the face of ‘high levels of uncertainty’, without quantifying ‘high’ either in general, or in the case of their specific application.

This is a common criticism of the given rationale for the application of T2 FSs and FLSs both from within and from beyond the FS research community. Consequently, it is important to investigate carefully what level of uncertainty warrants the application of T2 FSs and how this level can be precisely quantified. In other words, while it has been shown that IT2 FLSs can provide an improvement in terms of their ability to handle ‘more uncertainty’ than their T1 counterparts [38],[39], no systematic way of determining the potential advantage of employing T2 FSs over T1 FSs has been developed so far.

In the T2 literature it is common for the use of T1 FSs to be recommended when the level of uncertainty is low. When the uncertainty situation is medium and/or high, IT2 FSs are recommended. As noted, this is usually done without a clear quantification of the uncertainty qualifiers (i.e., low, medium or high) and without specifying the

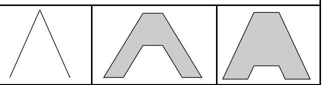
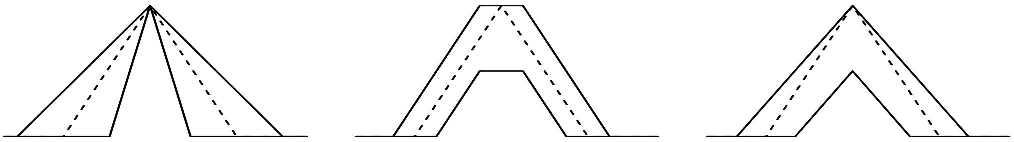
	FOU Size		
			
Uncertainty level			
Low	L		
Medium		M	
High			H

Figure 1.1 An illustration of uncertainty/FOU size relationships

actual FOU size appropriate for the given level of uncertainty (e.g., see Figure 1.1 for an illustration).

More to the point, even though it is well known that IT2 FSs reduce to T1 FSs when all uncertainty disappears (and as such their FOU) (e.g., in Figure 1.2(a), if the uncertainties about the left and right end points disappear, then the dashed triangular T1 MF remains.), the question of ‘T2 or T1?’ is often considered binary, even though intuitively the answer is continuous in nature, i.e., as the amount of uncertainty increases, the FOU of the FSs grows from initially T1 FSs to wider and wider IT2 FSs. This suggests the more appropriate question is, ‘How wide should the FOUs of the employed FSs be?’, rather than ‘should T1 or IT2 FSs be used?’ In this thesis, we address this challenging question by investigating the relationship between the FOU size of FSs and the level of uncertainty.

To explain the context and focus of this thesis, a number of potential methods for the creation of an IT2 FS by expanding an FOU around a principal T1 MF are shown in Figure 1.2. Figure 1.2(a) shows a standard blurring (SB) method which preserves the original T1 MF, but for which the size of the FOU (i.e. the width of a vertical slice for a given x) does not remain constant. This is unintuitive in many applications, in particular when the mound of uncertainty for a given variable (e.g., a sensor) is known in advance. We may not expect that the uncertainty varies in the degree of membership, certainly not as strongly as is shown in Figure 1.2(a). Figure 1.2(b)



(a) (b) (c)

Figure 1.2 Different IT2 FSs around a T1 principal MF (dashed). (a) Conventional IT2 FSs created using SB. (b) IT2 FS created using CB. (c) IT2 FS created using SP-CB.

shows a controlled blurring (CB) approach which preserves a uniform/constant level of uncertainty throughout the support of the lower membership function (LMF) of the FS, but does not maintain the shape of the original T1 MF (triangular in this case). Finally, in Figure 1.2(c), a shape-preserving controlled blurring (SP-CB) approach enables both constant uncertainty in memberships over the LMF and at the same time, the LMF and the upper membership function (UMF) keep their original T1 MF shape (a dashed line). Thus, the latter approach (SP-CB) enables both the adaptation of the IT2 FS for known levels of uncertainty (i.e., by increasing FOU size with increasing uncertainty) and the systematic comparison to the original (in this case triangular) T1 FLS to the resulting new IT2 FLS(s). This thesis adopts and details the CB and the SP-CB approaches and highlights the potential advantages and disadvantages of each approach in theoretical work and applications. A detailed investigation of the ability of both T1 FLSs and IT2 FLSs to model different levels of uncertainty/noise is conducted.

1.2 Aim and Objectives

The main aim of this thesis is to answer the following research question: 'What is the relationship between the size of the FOU of an IT2 FS and the uncertainty levels in a given application and how can understanding this relationship inform the design of FLSs?'

According to this aim, the research objectives are as follows:

1. To develop a process that enables the systematic transition from T1 to IT2 FSs in order to enable the comparison of FLSs which are equivalent in all but the size of the FOUs of the employed FSs. This will enable the systematic relating of FOU size to levels of uncertainty in a given application and vice versa.

2. To develop a comprehensive FLS design and evaluation process that describes the initial design of the T1 FLSs for a given application and its subsequent transformation into one or more IT2 FLSs under different levels of uncertainty.

3. To develop an analytical method that enables the selection of the FOU size in response to a known quantification of uncertainty in commonly used linguistic terms such as 'low', 'medium', and 'high'. This will provide an approach to develop FS based models that quantify such linguistic terms.

4. To refine the FSs and select the optimal FOU size in a given application faced with varying levels of uncertainty in order to enable better adaptation in real world applications by utilising the optimisation of IT2 FLSs.

5. To conduct a set of experiments to comprehensively evaluate the FLSs' performance using the developed methodologies.

1.3 Contributions Overview

This section provides a general overview of the thesis contributions. In this thesis, the uncertainty indicator is presented in order to study the uncertainty modelling of IT2 FSs. The uncertainty indicator is intended to show the amount of uncertainty captured by FSs and modelled by their FOUs. Specifically, the uncertainty indicator is the size of the primary membership interval at a given value, e.g., x for a given IT2 FS. With an assumption that a specific and constant level of noise/uncertainty associated with a given variable (e.g., level of measurement imprecision) is known and modelled by an IT2 FS, then, this FS should reflect the present level of uncertainty throughout its membership as far as possible (i.e., excluding its edges). Therefore, an FOU creation method should result to an equal amount of uncertainty to be captured over the primary domain of the MF, at least within the “core” of this FS (i.e., the support of the LMF). To achieve this, we propose two design methodologies that systematically vary (blur) the size of the FOU of IT2 FSs, enabling the systematic relating of FOU size to levels of uncertainty and vice versa. The two approaches are developed to create an FOU of an IT2 FS from an original T1 FS by transitioning from T1 to IT2 FSs for different levels of uncertainty (noise).

The first proposed method is a controlled blurring method used to create an FOU of an IT2 FS from an original T1 FS and maintain the level of uncertainty captured in the primary memberships of the FS constant, but not maintain the original T1 MF shape. The second proposed method is a shape-preserving controlled blurring method that follows a similar methodology as the first approach and also keeps the IT2 FS shape (LMF and UMF) the same as the original T1 MF. These two features are important as they enable the systematic relating of FOU size to levels of uncertainty and vice versa and enable an intuitive comparison of the T1 and T2 FSs.

Through a well-defined sample application domain that contrasted the performance of these resulting FLSs in the face of inputs with varying signal-to-noise ratios (SNRs) in a rich set of time series prediction (TSP) experiments, a distinct pattern of performance arising from the different levels-of-uncertainty and FOU-size combinations is explored and captured.

It is worth noting that the main objective of this thesis is not to achieve optimal performance in a given application (such as a TSP), but to provides the optimal size of the FOUs (from T1 to T2) for given levels of uncertainty/noise. This process will provide significant insight into the relationship levels/amount of uncertainty and the appropriate FOU size in a given domain.

Based on this, we provide a guidelines for the selection of appropriate FOU sizes for given levels of uncertainty and propose an approach to quantifying the commonly used linguistic labels such as 'low', 'medium' and 'high' through FS models.

To refine the FSs for existing systems and select the optimal FOU size in a given application faced with varying levels of uncertainty in order to enable a better adaptation in real world applications, an optimised controlled blurring method used for optimising IT2 FLS, in which the controlled blurring method is adopted. Then, the design parameters are tuned through an optimisation method. The controlled blurring approach has few parameters to be tuned as only a single extra parameter is added to a T1 MF that is used to define IT2 MFs. We demonstrate the approach through application to a TSP problem, using training data sets corrupted with different levels of noise.

1.4 Organisation of the Thesis

The thesis is organised as follows:

- Chapter 2 provides the essential background and an overview of the relevant literature. It provides a detailed discussion of fuzzy logic sets and systems including T1 and T2 FSs and systems. In addition, TSP as an application and FLS optimisation and learning techniques are presented.

- Chapter 3 is pivotal for the argument of the whole thesis. It provides details on the main investigative framework that covers the detailed analysis and presentations of the FOU creation methodologies and their empirical evaluation. It also discusses the potential of a large number of T2 FS generation algorithms and outlines the subset of three specific cases addressed in this thesis.

- Chapter 4 provides an investigation into the controlled blurring approach for the creation of IT2 FSs and associated set of experiments to explore the relationship between FOU size and uncertainty levels.

- Chapter 5 provides an investigation into the shape-preserving controlled blurring approach with different properties, in combination with a comprehensive set of experiments to study another FOU creation method using different configurations, such as training FLSs using training data corrupted with different noise levels and singleton and non-singleton fuzzifications.

- Chapter 6 provides an exploration into the optimised controlled blurring approach, particularly focusing on the controlled blurring approach by utilising the simulated annealing algorithm, to refine the FSs for existing systems and select the optimal FOU size in a given application faced with varying levels of uncertainty in order to better adapt in real world applications.

- Chapter 7 gives the steps used to quantify and model linguistically expressed uncertainty levels such as ‘low, medium, high’ that are commonly used in the fuzzy systems field to describe the levels of uncertainty in given applications.

- Chapter 8 draws conclusions, lists the contributions arising from this work, mentions the limitations and suggests some interesting potential directions for future research arising from this thesis.

Chapter 2

Literature Review

2.1 Introduction

This chapter presents the necessary foundation for the presentation and definitions of the methodology used in the following chapters and the application of the investigative approach presented later in thesis.

As stated, the focus of this thesis is to study the behaviour of FLSs with different FOU sizes (from type-1 to type-2) exposed to different, well defined uncertainty/noise. This process will provide significant insight into the relationship between levels/amount of uncertainty and appropriate FOU size.

We first introduce in this chapter fuzzy systems and their ability to model and handle uncertainty showing the different uncertainty models of the three common FSs, i.e. T1, IT2 and GT2. Then we provide a review of T1 and IT2 FLSs.

This chapter will also present TSP as a well-defined sample application.

Learning and optimisation of FLSs are two different methods that are employed. In FLS learning problems, an approach used to generate the system rule base and its parameters that define the FSs from a given input-output data is performed. Conversely, in optimisation problems, a predefined rule base and initial system parameters are used and the objective is to search for and find a better set of parameters defining the FLSs. Both of these methods will be generally presented in this chapter as their concepts are used in this thesis.

The rest of this chapter is organised as follows: Section 2.2 introduces fuzzy systems and uncertainty followed by Section 2.3 that covers type-1 fuzzy sets. Section 2.4 introduces the general concepts of type-1 fuzzy logic systems. Section 2.5 is dedicated to reviewing type-2 fuzzy sets followed by the general concepts of type-2 fuzzy logic systems in Section 2.6. In Section 2.7, transitioning from type-1 to type-2 fuzzy sets and systems is detailed. Section 2.8 introduces time series prediction as an application area followed by Section 2.9 that presents the learning and optimisation of fuzzy logic systems. Finally, Section 2.10, provides the chapter summary.

2.2 Fuzzy Systems and Uncertainty

Fuzzy set theory was first introduced by Zadeh [1] in 1965. The field of FSs has evolved over the last 50 years and FSs have been accepted as an adequate methodology for developing systems that are able to deliver adequate performance in the face of uncertainty and imprecision [39–42]. Hence, FLSs have been successfully implemented in many real world applications and are used in many areas. In addition, FS theory provides a simple and efficient method of designing FLSs that is close to human thinking and perception [43].

In particular, fuzzy logic control (FLC), as one of the earliest applications of FLSs, has become one of the most successful applications. The first FLC was developed in

1975 by Mamdani and Assilian to control a steam engine in a laboratory [44]. The first industrial application of fuzzy logic was developed in 1976 by Blue Circle Cement and SIRA in Denmark for a cement kiln controller that went into operation in 1982 [45]. In 1987, the Sendai subway system in Japan operated an automatic train controller based on fuzzy logic systems. Since then, FLSs have been applied with great success to many applications. Recently, FLSs have been used in many real-world applications used in people's daily lives such as washing machines, air conditioners, video cameras, medical diagnosis systems, car braking systems, etc. [6],[40],[46].

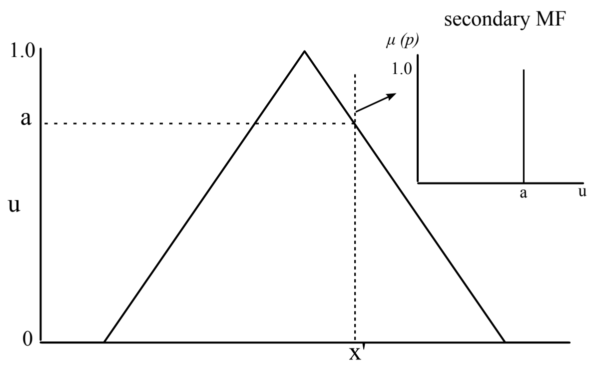
FLSs are widely accepted for their ability to model and handle uncertainty. Several efforts have been made to define the uncertainties in FLSs and their underlying sets. A general discussion about uncertainty was presented by Klir and Weirman [47] showing three types of uncertainty: fuzziness, strife and non-specificity. Fuzziness (vagueness) results from the imprecise boundaries of FSs [13]. Non-specificity (imprecision) is linked to information-based imprecision. Strife (discord) expresses conflicts among the various sets of alternatives [13].

The T1 FLS is the most common and widely used FLS. T1 FLSs are based on T1 FSs and have been successfully applied in many applications, such as control systems (especially for the control of complex non-linear systems that are difficult to model analytically [48],[49]), decision making, classification problems, system modelling and others [50]. However, it has been shown that there are limitations in the ability of T1 FSs to directly model and minimise the effects of uncertainty [51], [3], [13]. This is because T1 MFs are precise [13] and their membership grade is a crisp value (see Figure 2.1(a)).

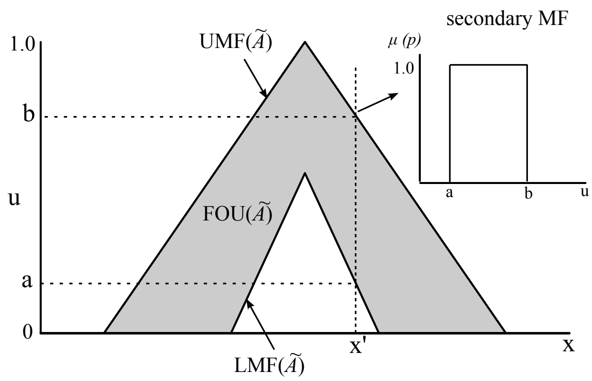
Recently, a significant increase in research has been devoted to more complex forms of fuzzy logic such as IT2 FLSs and, more recently, GT2 FLSs. This advancement is due to the realisation that T1 FSs can only handle a limited range of uncertainty whereas T2 FSs allow for better modelling of uncertainty as they encompass an FOU, which, associated with their third dimension, gives more degrees of freedom to the use of T2 FSs in comparison to T1 FSs (i.e., T2 FSs are described by MFs that are characterized by more parameters than are MFs for T1 FSs [16]).

T2 FSs was introduced by Zadeh [14] in 1975 as an extension of the concept of T1 FSs, characterised by MFs that are themselves fuzzy for which the membership degrees are expressed as FSs on [0,1], have been widely accepted as more capable of modelling higher orders of uncertainty than T1 FSs [4],[16, 52–55]. Mendel [13] argued that T1 FSs are inadequate to model many types of uncertainty that can be present in an FLS, whereas T2 FSs are able to handle them including the three types of uncertainty : fuzziness, strife, and non-specificity. IT2 FSs [13] are a special case of GT2 FSs and currently the most widely used due to their great reduction in computational cost. It has been shown in the literature many types of FSs such as interval-valued FSs, gray set, intuitionistic FSs, etc. [56],[57]. For more details about the definitions and the relationships between different FSs can be found in [58]. IT2 FSs will be the focus of this thesis. There is a rapidly growing field of research on IT2 FLSs with much associated evidence of successful applications [4].

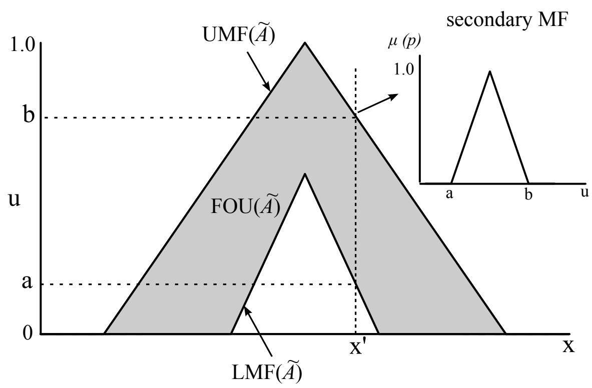
Figure 2.1 shows the three different types of fuzzy sets. A T1 FS shown in Figure 2.1(a), an IT2 FS shown in Figure 2.1(b) and a GT2 FS shown in Figure 2.1(c). These figures also show the secondary MFs (third dimension) of the three fuzzy sets using the same input $x = x'$. It is important to consider the uncertainty models provided



(a) Type-1 fuzzy set



(b) Interval type-2 fuzzy set



(c) General type-2 fuzzy set

Figure 2.1 An example of the three types of FSs with view of the secondary MFs (defined in Section 2.5.1) with same input x' (a) type-1 fuzzy set, (b) interval type-2 fuzzy set and (c) general type-2 fuzzy set.

by these three types of FSs. In the T1 FS, the degrees of membership are specified as crisp numbers taking values within the interval [0,1]. In the GT2 FS, the degrees of membership are themselves fuzzy and each is specified as a T1 FS (a secondary membership function). In a case when the secondary MF is equal to 1, then GT2 FS will reduce to an IT2 FS. The T1 FS is depicted in Figure 2.1(a), showing the value of the primary domain (membership) at $x = x'$ that takes only one value, a, at which the secondary grade equals 1. So, in the case of the T1 FS, there is no uncertainty associated with the primary membership value at each x value [13]. For an IT2 FS (as shown in Figure 2.1(b)), the primary domain (membership) at $x = x'$ takes values within the interval [a,b] and each point in this interval has a secondary membership equal to 1. Hence, an IT2 FS contains a maximum amount of uncertainty that is equally distributed over the interval [a,b] [59]. For a GT2 FS (as shown in Figure 2.1(c)), the primary (domain) membership at $x = x'$ also takes values within the interval [a,b], but is different from the IT2 FS since each point in this interval has a different secondary membership. Overall, it can be observed that the uncertainty that is associated with a GT2 FS is located between the uncertainty of a T1 FS and an IT2 FS [59].

While this thesis focuses on IT2 FLSs, it also provides background material about T1 FLSs. Thus, before we can explain what T2 FSs and systems are, we must explain what T1 FSs and T1 FLSs are because T2 FSs build upon T1 FSs.

2.3 Type-1 Fuzzy Sets

A fuzzy set was defined originally by Zadeh [1] as an extension of a (classical) crisp set. In crisp sets, the membership of elements is a binary value, which allows an element to either belong (full membership) or not belong to the set (no membership at all). In contrast, FSs permit partial membership so that an element may partially belong to a set. In a crisp set, the membership or non-membership of an element x in the crisp set A is described by the membership function (MF) (also called characteristic function) μ_A of A, where [13],[60],[61]

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

Fuzzy set theory extends this concept by defining partial memberships, which can take values in the interval [0,1]. In a T1 FS (see Figure 2.1(a)) also called simply a FS, A is defined on a universe of discourse X and the MF for A is $\mu_A \colon X \to [0,1]$ [13]. For each element in the set, $x \in X$, the value of $\mu_A(x)$ is the degree of membership of x in A between zero and one. A fuzzy set A in universe of discourse X can be represented in pairs of x and the value of its MF, $\mu_A(x)$ as [13]:

$$A = \{(x, \mu_A(x)) \mid \forall x \in X\} \tag{2.2}$$

If the fuzzy set A has a continuous universe of discourse X, it can be written as [1],

$$A = \int_X \mu_A(x)/x, \tag{2.3}$$

where the integral sign denotes union, and the division sign represents association (the collection of all points $x \in X$ with associated MF $\mu_A(x)$). If the fuzzy set A has a discrete universe of discourse X, it can be written as [1],

$$A = \sum_X \mu_A(x)/x, \tag{2.4}$$

where the summation sign denotes the set-theoretic operation of union and the division (slash) sign represents association (the collection of all points $x \in X$ with associated MF $\mu_A(x)$).

2.3.1 Properties of Type-1 Fuzzy Sets

To describe T1 FSs more specifically, we will define some of their properties [14],[62],[63].

The 'support' of a T1 FS A defined on X is a crisp set that contains all the elements of X ($x \in X$) that have non-zero membership grades in A (i.e., $\mu_A(x) > 0$) and is defined as:

$$\text{support}(A) = \{x \in X \mid \mu_A(x) > 0\} \tag{2.5}$$

A T1 FS that has a single point in X as the support with $\mu_A(x) = 1$ is called a 'singleton' T1 fuzzy set (see Figure 2.3(a)).

The 'core' of a fuzzy set A is the set of all points $x \in X$ such that $\mu_A(x) = 1$:

$$\text{core}(A) = \{x \in X \mid \mu_A(x) = 1\} \tag{2.6}$$

The 'height' of T1 FS A is the largest membership value, and it is defined as follows:

$$\text{height}(A) = \text{supp}_{x \in X}(\mu_A(x)) \tag{2.7}$$

A fuzzy set A is called 'normal' when height(A) = 1 and it is called 'subnormal' when height(A) < 1.

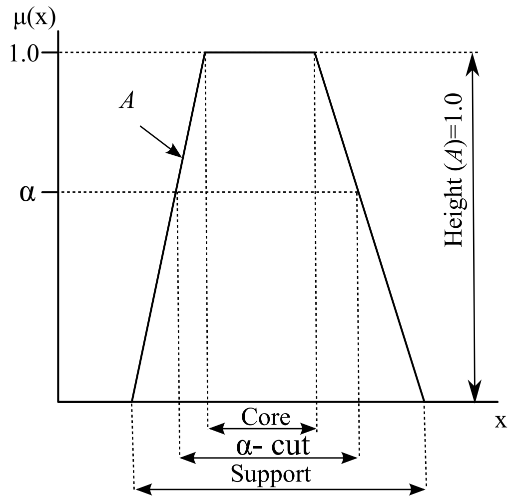


Figure 2.2 An example of a trapezoidal membership function and some of its properties

One of the most important concepts of fuzzy sets is the concept of an α-cut. The α-cut of an FS A is the 'crisp' set of all elements that have a membership value greater than or equal to α. A T1 FS A is defined on X and α is a number in [0,1], an α-cut, A_α is defined by [61]:

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\} \tag{2.8}$$

These properties are illustrated by the trapezoidal membership function A in Figure 2.2

The membership functions commonly used in practice are singleton, triangular, trapezoidal, Gaussian, and bell-shaped [13]. Generally, MFs can either be chosen by experts (users) or they can be created using optimisation methods [64],[65],[66]. Examples of singleton, triangle, trapezoidal and Gaussian MFs are depicted in Figure 2.3.

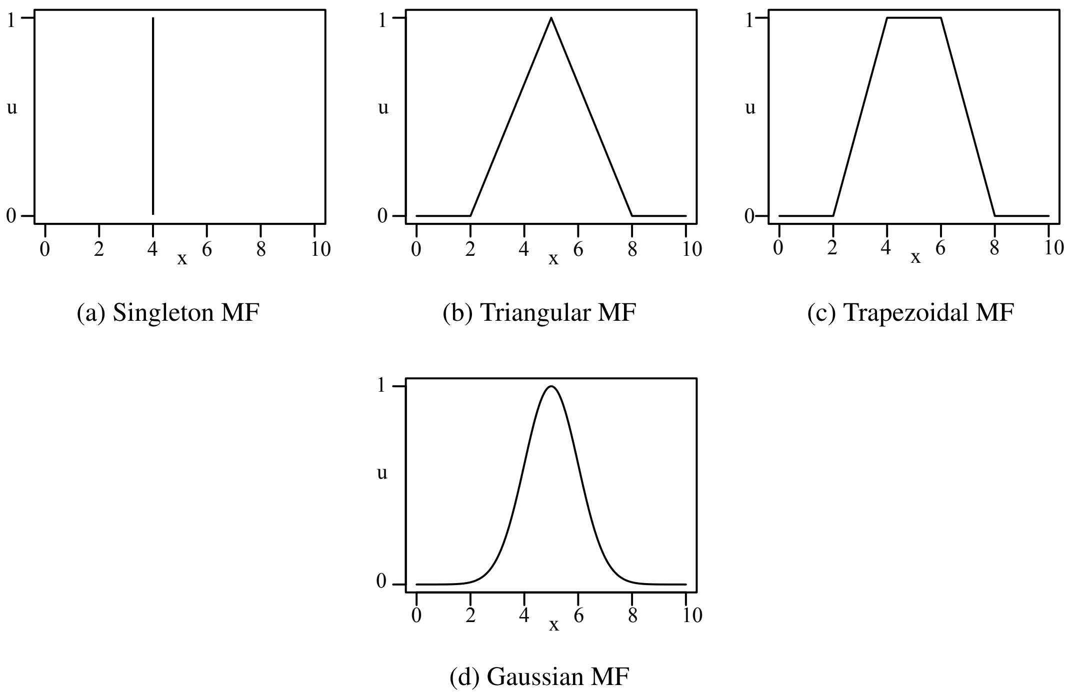


Figure 2.3 Examples of four T1 MFs (a) Singleton MF, (b) triangular MF, (c) trapezoidal MF and (d) Gaussian MF.

2.3.2 Type-1 Fuzzy Set Operations

Corresponding to the crisp set operations of union, intersection and complement, fuzzy sets have the same operations and are called Zadeh's operations as they were initially defined in Zadeh's paper [1].

Assume T1 FSs A and B are two subsets of X and are defined by their MFs $\mu_A(x)$ and $\mu_B(x)$. The union of A and B is defined by $\mu_{A\cup B}(x)$:

$$\mu_{A\cup B}(x) = \max[\mu_A(x), \mu_B(x)], \quad \forall x \in X \tag{2.9}$$

The intersection of A and B is defined by $\mu_{A\cap B}(x)$:

$$\mu_{A\cap B}(x) = \min[\mu_A(x), \mu_B(x)], \quad \forall x \in X \tag{2.10}$$

The product of A and B is defined by $\mu_{A*B}(x)$:

$$\mu_{A*B}(x) = \text{prod}[\mu_A(x), \mu_B(x)], \quad \forall x \in X \tag{2.11}$$

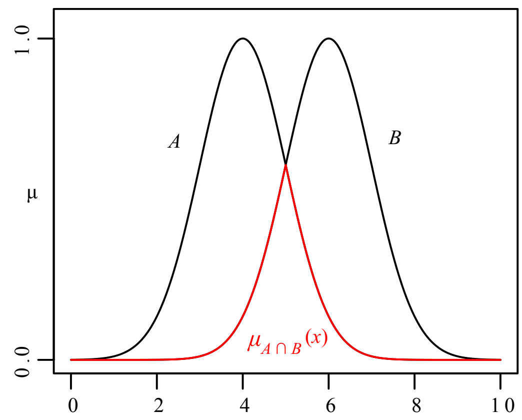
The complement of A is defined by $\mu_{A'}(x)$:

$$\mu_{A'}(x) = 1 - \mu_A(x), \quad \forall x \in X \tag{2.12}$$

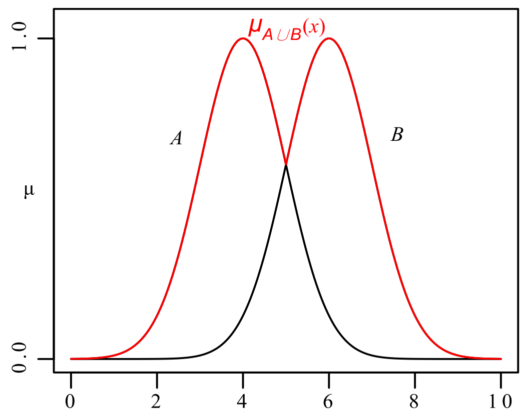
Thus the intersection, product and union operations are used to combine T1 FSs. They are equivalent to the operators 'AND' and 'OR' used in classical logic. Examples of the union, intersection, product and complement of the two T1 FSs A and B using (2.9), (2.10), (2.11) and (2.12) are depicted in Figure 2.4. While, different t-norms and t-conorms have appeared in the literature [67–69] and were developed to generally define the operations, $\mu_{A\cup B}(x)$ and $\mu_{A\cap B}(x)$, in this thesis only product t-norm (2.11) that is used by the inference engine to combine the firing strengths from multiple antecedents and the maximum t-conorm (2.9) that is used to combined output fuzzy subset by taking the maximum (union) over all of the fuzzy subsets.

2.4 Type-1 Fuzzy Logic Systems

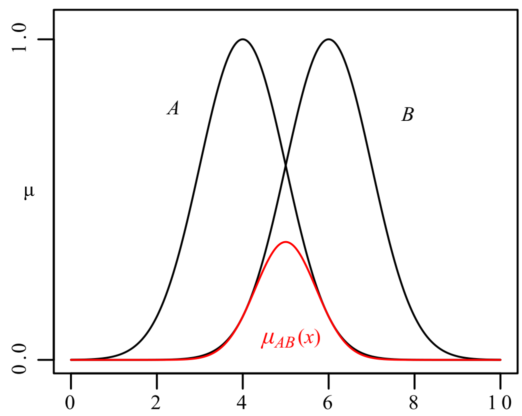
T1 FLSs are also known as fuzzy rule-based systems that can be considered as systems that map crisp inputs into outputs by utilising fuzzy sets theory [70]. The main types of FLSs are Mamdani [71] and Takagi & Sugeno [72]. In this thesis, the Mamdani method is used because the core part of our investigations has been trying to capture the linguistic uncertainty, thus, using FSs in the whole FLS (including in the output



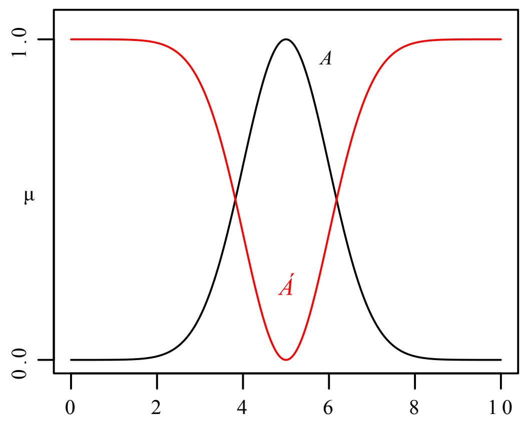
(a) Intersection of two fuzzy sets



(b) Union of two fuzzy sets



(c) Product of two fuzzy sets



(d) Complement of a fuzzy set

Figure 2.4 Examples of fuzzy operations using two T1 FSs *A* and *B*. (a) Intersection of two FSs, (b) union of two FSs,(c) product of two FSs and (d) complement of an FS.

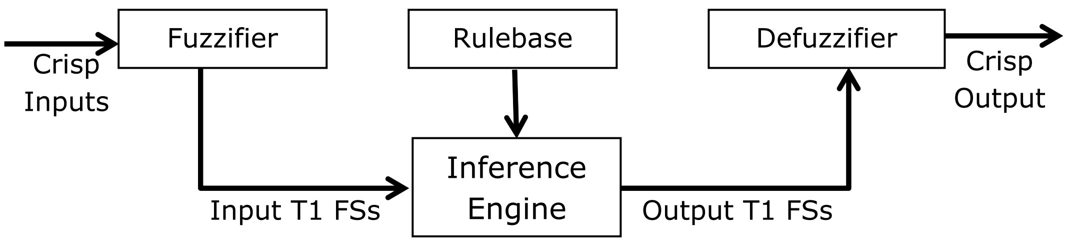
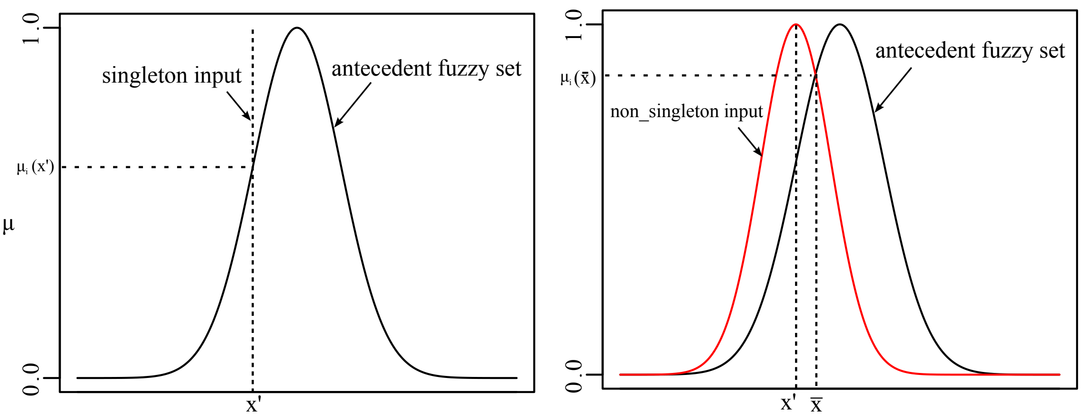


Figure 2.5 A type-1 fuzzy logic system.

modelling) is intuitive and valuable. In the also popular Takagi & Sugeno FLSs, the output is characterised by a function, rather than a FS, thus it is not couched in the context of using linguistic terms. The well-known Mamdani fuzzy model contains four components: fuzzifier, rule base, inference engine and defuzzifier [13]. Figure 2.5 shows these components. As shown, crisp inputs are first fuzzified into type-1 fuzzy input sets. These activate the inference engine and the rule base to produce output T1 FSs, which are then combined to produce an aggregated T1 output FS. Finally, a defuzzifier produces a crisp output from the fuzzy output set(s) resulting from the inference engine. Further details on T1 FLSs can be found in [13],[73]. The background and description of each of these components in the context of T1 FLSs are summarised below.

2.4.1 Fuzzifier

The fuzzifier maps crisp inputs into a membership grade of one or more T1 FSs, based on the given membership functions. This is achieved by evaluating the crisp inputs and assigning each input a membership degree $\mu_A(x)$ in its input FS. According to the type of fuzzification [13], T1 FLSs can be divided into singleton fuzzy logic systems (SFLSs) and non-singleton fuzzy logic systems (NSFLSs). A non-singleton T1 FLS has the same structure as a singleton T1 FLS, and they share the same type of rules; the only difference is the type of fuzzification. The majority of FLSs are using SFLS because singleton fuzzification is simpler and faster to compute. In singleton fuzzification, inputs are considered to be singleton FSs (see Figure 2.3), while non-singleton fuzzification models the FLS inputs as FSs. Non-singleton fuzzification allows better modelling of input uncertainties by modelling inputs as fuzzy sets and



(a) Singleton fuzzification

(b) Non-singleton fuzzification

Figure 2.6 Examples of the fuzzification of a crisp input using (a) singleton and (b) non-singleton fuzzification.

modelling linguistic uncertainty using antecedent FSs in two steps [74]. To better cope with noisy and imprecise input measurements, the non-singleton fuzzifier is an efficient choice [75]. Figure 2.6 shows singleton and non-singleton fuzzification. More details on non-singleton fuzzification can be found in [13],[76–78]. A rich discussion on fuzzification, membership function creation and desired output can be found in [79].

2.4.2 Rule Base

Fuzzy rule base (set of fuzzy rules) is the core part of an FLS that is used in the inference process. The most commonly used are: Mamdani [71], where the rule consequents are FSs, and Takagi & Sugeno [72], where the rule consequents are crisp functions of the inputs. The rules can be expressed as a collection of conditional statements in the form IF–THEN statements. Without a loss of generality, the multiple inputs single output (MISO) system with the output variable y is considered here. In

Mamdani-type fuzzy rules (first used by Mamdani in 1977), each rule is in the form of (2.13) with n inputs $x_1 \in X_1, \dots, x_n \in X_n$ and one output $y \in Y$.

$$\text{IF } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \ \dots \text{ and } x_n \text{ is } A_n \text{ THEN } y \text{ is } B, \tag{2.13}$$

where $x_1, x_2, \dots, x_n$ are the input variables to the FLSs and $A_1, A_2, \dots, A_n$ are the input FSs in the antecedent part and y is the output variable and B is the output FSs.

A fuzzy rule contains two parts, the IF part called the antecedent part and the THEN part called the consequent part. To generate these rules, many methods can be used such as deriving them from experts or from given numerical data sets as shown by [10],[80],[81] where they generated fuzzy rules from training data. FSs are associated with the terms shown in the antecedent or consequent parts of rules and MFs are used to describe these FSs [13].

2.4.3 Inference Engine

The inference engine in the Mamdani type (Equation (2.14)) provides a mapping of input FSs to output FSs using the rules (see (2.13)) from the rule base and the connecting operators such as the union (OR in classical logic) or the intersection (AND in classical logic). The result is an output fuzzy set by using (2.14), then the defuzzifier converts them to a crisp output. In this thesis, the inference system uses the AND operator for connecting inputs. The Mamdani implication (inference) [13]:

$$\mu_{A^* \to B}(x^{**}, y) \equiv \mu_{A^*}(x^{**}) \bigstar \mu_B(y), \tag{2.14}$$

where $\bigstar$ is product or minimum operation, $A^* \in A_1, A_2, \ldots, A_n$, $x^{**} \in x_1, x_2, \ldots, x_n$, $\mu_{A^*}(x^{**})$ is the input MF and $\mu_B(y)$ is the consequent MF.

2.4.4 Defuzzifier

Once the inference system has produced output fuzzy sets, the defuzzifier in the Mamdani model converts them into crisp values. In general, there are many methods (defuzzifiers) that have been proposed in the literature for the defuzzification of type-1 fuzzy sets such as centroid (also known as centre of gravity (COG)), centre of sets (COS), height and centre of sums [13]. The centroid method is one of the most commonly used defuzzification strategies. It returns the COG value and it is the one used in this thesis. The process to find the crisp output using $y_{COG}(X)$ is defined using (2.15) [13]:

$$y_{COG}(X) = \frac{\sum_{i=1}^{N} y_i \mu_B(y_i)}{\sum_{i=1}^{N} \mu_B(y_i)}, \tag{2.15}$$

where $\mu_B(y_i)$ is the aggregated value of $\mu_B(y)$, N is the number of discretized points that are used to find the COG of B and y_i is the output variable. For more complete details, sources such as [13],[82],[83] have provided comparisons of different methods of defuzzification.

2.5 Type-2 Fuzzy Sets

2.5.1 General Type-2 Fuzzy Sets

The concept of GT2 FSs was introduced by Zadeh in 1975 [14] as an extension of the concept of T1 FSs. A T2 FS is characterised by a fuzzy MF that has a membership grade for each element of the set as a fuzzy set in [0,1], whereas, a T1 FS has a membership grade with a crisp value in [0,1]. T2 FSs also known as fuzzy valued fuzzy sets or fuzzy-fuzzy sets [84] referring to their MF having fuzzy values.

A T2 FS also known as a GT2 FS (see Figure 2.1(c)), $\tilde{A}$, may be represented as [18]:

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) \mid x \in X, u \in J_x \subseteq [0,1]\}, \tag{2.16}$$

where J_x is the primary membership of x and $\mu_{\tilde{A}}(x,u) \in [0,1]$ is the secondary MF. $\tilde{A}$ in (2.16) can also be defined as:

$$\tilde{A} = \int_{x\in X} \int_{u\in J_x} \mu_{\tilde{A}}(x,u)/(x,u),\ J_x \subseteq [0,1], \tag{2.17}$$

where $\int$ denotes union. For discrete universes of discourse, set $\tilde{A}$ is defined as:

$$\tilde{A} = \sum_{x\in X} \sum_{u\in J_x} \mu_{\tilde{A}}(x,u)/(x,u),\ J_x \subseteq [0,1], \tag{2.18}$$

GT2 FSs have limited use in practice due to the significant increase in computational cost during their implementation processes. However, in recent years, some advances in GT2 FLSs research including faster type-reduction methods are achieved enabling more practicality. Gafa and Coupland [85] introduced a recursive algorithm to reduce some of the computations complexity of GT2 type-reduction. More practical type-reducers were proposed including Coupland and John [19] where they introduced the geometric defuzzifier based on the geometric representation, the sampling algorithm proposed by Greenfield et al. [26], which uses a random samples of embedded sets to find an approximation to the exact value and the importance sampling defuzzifier introduced by Linda and Manic [86].

The recent developments of T2 FSs theory and applications have shown for example how to model interval-based data using GT2 FSs based on zSlices [87]. It has also been shown that the GT2 FLSs that employed GT2 FSs outperform their T1 and IT2 counterparts in case of the presence of perturbations [88] and when the uncertainties at different levels increase [89].

2.5.2 Interval Type-2 Fuzzy Sets

IT2 FSs (see Figure 2.1(b)) are characterised by their secondary MFs which only take the values of one or zero. This reduction greatly simplifies the computational complexity required in processing GT2 FSs [13],[90],[27]. When all the secondary grades $\mu_{\tilde{A}}(x,u)$ are equal to 0 or 1, then, $\tilde{A}$ is an IT2 FS. Following (2.17), an IT2 FS $\tilde{A}$ can be expressed as [13]:

$$\tilde{A} = \int_{x\in X} \int_{u\in J_x} 1/(x,u) \; J_x \subseteq [0,1] \tag{2.19}$$

IT2 FSs are easier to compute than GT2 FSs and this is the main driver for their wide usage in real world applications (e.g., in control [91] and in the supplier selection problem [92]).

2.5.3 Footprint of Uncertainty

The term 'footprint of uncertainty' (FOU) (see Figure 2.1(b)) provides a very useful verbal and graphical representation of the uncertainty in the primary memberships of an IT2 FS [13]. The domain of the primary membership J_x defines the FOU (the union of all primary memberships) of FS $\tilde{A}$ [13]:

$$FOU(\tilde{A}) = \bigcup_{\forall x \in X} J_x \tag{2.20}$$

The lower membership function (LMF) is denoted $\underline{\mu}_{\tilde{A}}(x)$ (or LMF($\tilde{A}$) and the upper membership function (UMF) is denoted $\overline{\mu}_{\tilde{A}}(x)$ (or UMF($\tilde{A}$) of $\tilde{A}$ are two T1 MFs that bound $FOU(\tilde{A})$ (e.g., see Figure 2.1(b)). The support of LMF($\tilde{A}$) is the crisp set of all points such that LMF($\tilde{A}$)>0 and the support of UMF($\tilde{A}$) is the crisp set of all points such that UMF($\tilde{A}$)>0. Alternatively, the FOU of an IT2 FS $\tilde{A}$ can be described by its lower LMF and UMF [15]:

$$FOU(\tilde{A}) = \bigcup_{\forall x \in X} [\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x)] \tag{2.21}$$

As an IT2 set is completely defined by its FOU [93], i.e., its LMF and UMF, computation is drastically simplified in comparison to GT2 FSs.

Mendel and John [94] introduced the concept of the domain of uncertainty (DOU) for a T2 FS, $\tilde{A}$, as:

$$DOU(\tilde{A}) = \bigcup_{\forall x \in X} J_x \tag{2.22}$$

Here, the DOU is the union of all the primary memberships of $\tilde{A}$. They developed this concept to differentiate between two different primary variables, namely, 'naturally ordered' variables (e.g., temperature, pressure, etc.) and not 'naturally ordered' variables (e.g., beautiful, happy, etc.). This concept also helps distinguish T2 FSs as discrete, continuous or hybrid. For continuous T2 FSs with a 'naturally ordered' primary variable, as in this thesis, the DOU is equal to the FOU, i.e., $DOU(\tilde{A}) = FOU(\tilde{A})$ [94]. However, since the term FOU is already firmly established in the T2 literature, we will continue to use it in this thesis.

2.5.4 Interval Type-2 Fuzzy Set Operations

The following is a summary of the set-theoretic operations for the union and intersection of two IT2 FSs and also the complement of an IT2 FS because these operations are widely used in IT2 FLSs.

The union of two IT2 FSs, $\tilde{A}$ and $\tilde{B}$, is equivalent to the 'OR' operation in boolean and is defined by:

$$\tilde{A} \cup \tilde{B} = 1/[\underline{\mu}_{\tilde{A}}(x) \vee \underline{\mu}_{\tilde{B}}(x), \overline{\mu}_{\tilde{A}}(x) \vee \overline{\mu}_{\tilde{B}}(x)], \quad \forall x \in X \tag{2.23}$$

The intersection of two IT2 FSs, $\tilde{A}$ and $\tilde{B}$, is equivalent to the 'AND' operation in boolean and is defined by:

$$\tilde{A} \cap \tilde{B} = 1/[\underline{\mu}_{\tilde{A}}(x) \wedge \underline{\mu}_{\tilde{B}}(x), \overline{\mu}_{\tilde{A}}(x) \wedge \overline{\mu}_{\tilde{B}}(x)], \quad \forall x \in X \tag{2.24}$$

The complement of an IT2 FS $\tilde{A}$, denoted $\overline{\tilde{A}}$ is:

$$\overline{\tilde{A}} = 1/[1 - \overline{\mu}_{\tilde{A}}(x), 1 - \underline{\mu}_{\tilde{A}}(x)], \quad \forall x \in X \tag{2.25}$$

As shown, $\underline{\mu}_{\tilde{A}}(x)$ and $\underline{\mu}_{\tilde{B}}(x)$ are the lower MFs, $\overline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{B}}(x)$ are the upper MFs, $\vee$ is the max function and $\wedge$ is the min function.

2.6 Type-2 Fuzzy Logic Systems

The IT2 FLS is depicted in Figure 2.7 and has five components: fuzzifier, inference engine, rule base, type-reducer and defuzzifier. It is very similar to the structure of a T1 FLS with the only difference being the introduction of a type-reducer component [13]. An IT2 FLS employs IT2 FSs to represent its inputs and outputs. An FLS that uses at least one T2 FS (IT2 FS) is referred to as a T2 FLS (IT2 FLS). In a T2 FLS, crisp inputs are first fuzzified, most commonly into fuzzy singletons, resulting in so-called Singleton T2 FLSs. The fuzzified inputs activate the inference engine and the rule base to produce output T2 FSs, which are then combined to produce an aggregated T2 output FS. They are then processed by a type-reducer such as the centroid type-reducer, which performs a centroid calculation, leading to T1 FSs known as type-reduced sets [13]. The defuzzifier finally defuzzifies the type-reduced type-1 fuzzy outputs to produce crisp outputs. Further detail on T2 FLSs can be found for example in [13]. The background and description of each of these components in the context of IT2 FLSs are summarised below.

2.6.1 Fuzzifier

In fuzzification, the fuzzifier maps a crisp input into a membership grade of one or more IT2 FSs, based on the given membership functions (the antecedents part of the rules) using the upper $\overline{\mu}_{\tilde{A}}(x)$ and the lower $\underline{\mu}_{\tilde{A}}(x)$ membership functions and then finds

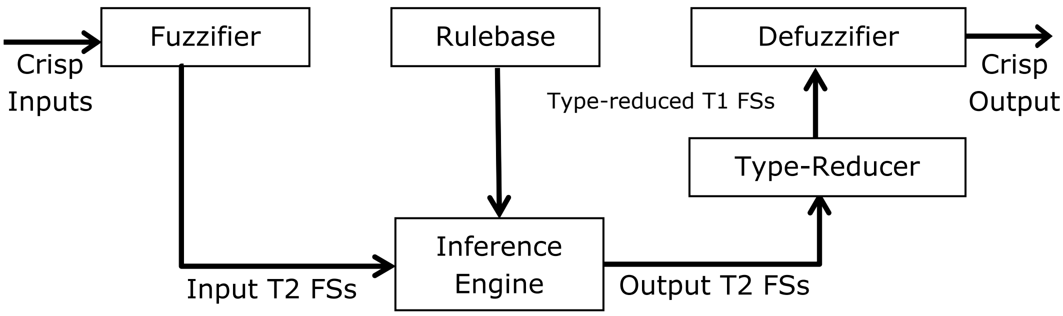


Figure 2.7 Interval type-2 fuzzy logic system.

an interval that will be used in the inference engine. In the case of interval type-2 fuzzy logic systems, the resulting fuzzified value is an interval type-1 fuzzy set.

2.6.2 Rule Base

General form of the rules of T1 FLSs are shown in (2.13). The difference between T1 and IT2 FLS rules is related to the nature of the membership functions. The structure of the rule base remains exactly the same in the IT2 FLS case, but now some or all of the FSs used in the system are IT2 FSs. Without the loss of generality the multiple inputs single output (MISO) system with output variable y is considered here. As for a T1 FLS, the IT2 FLS has n inputs $x_1 \in X_1, \ldots, x_n \in X_n$ and one output $y \in Y$ and each rule is of the form in (2.26).

$$\text{IF } x_1 \text{ is } \tilde{A}_1 \text{ and } x_2 \text{ is } \tilde{A}_2 \ \ldots \text{ and } x_n \text{ is } \tilde{A}_n \text{ THEN } y \text{ is } \tilde{B}, \tag{2.26}$$

where $x_1, x_2, \ldots, x_n$ are the input variables to the FLSs; $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_n$ are the input IT2 FSs in the antecedent part; y is the output variable and $\tilde{B}$ is the output IT2 FS. When some or all of the antecedent and consequent T2 FSs are IT2 FSs, then it is called an interval type-2 fuzzy logic system (IT2 FLS). These are the FLSs used in this thesis.

2.6.3 Inference Engine

The inference engine provides a mapping of input IT2 FSs to output IT2 FSs using the rules (see (2.26)) from the rule base and operators such as join ($\sqcup$) and meet ($\sqcap$) operators [18] which are used instead of union and intersection operators. Equations (2.27) and (2.28) are the meet and join operations for IT2 FSs. The result of their operations in inference engine is the output fuzzy sets that are passed into the type-reduction. The type-reducer transforms them into type-1 fuzzy reduced sets that can be defuzzified to a crisp output.

Let $F = \int_{\upsilon \in F} 1/\upsilon$ and $G = \int_{w \in G} 1/w$ be two interval sets with domains $\upsilon \in [l_f, r_f] \subseteq [0,1]$, and $g \in [l_g, r_g] \subseteq [0,1]$ respectively. Then, the meet ($\sqcap$) between F and G, $Q = F \sqcap G = \int_{\upsilon \in Q} 1/q$, under minimum or product is [29]:

$$Q = F \sqcap G = \int_{q \in [l_f * l_g, r_f * r_g]} 1/q, \tag{2.27}$$

where $q = \upsilon * w$. The join ($\sqcup$) between F and G, $Q = F \sqcup G = \int_{\upsilon \in Q} 1/q$, is given by [29]:

$$Q = F \sqcup G = \int_{q \in [l_f \vee l_g, r_f \vee r_g]} 1/q, \tag{2.28}$$

where $q = \upsilon \vee w$. From equations 2.27 and 2.28, the meet and join operation of interval sets are determined by $[l_f, r_f]$ and $[l_g, r_g]$ which are the two end-points of each interval set.

2.6.4 Type-Reducer

This component of an IT2 FLS is used to reduce IT2 FSs resulting from the inference engine to T1 FSs known as type-1 fuzzy reduced sets by performing one of the type-reduction methods. There are many type-reduction methods that have been reported in the literature such as centroid, center-of-sums, height, modified-height and center-of-sets type-reducers [95],[13]. As in T1 FLSs, where the center-of-sets defuzzification step drastically simplifies computation but sacrifices precision, in IT2 FLSs, the center-of-sets type reduction facilitates computation but omits detail. As this thesis focusses on the detailed modelling of uncertainty rather than performance, we thus choose centroid type-reduction.

The most widely used algorithm for performing centroid type-reduction is the Karnik-Mendel algorithm (KM) [95]. An improved version of KM has also been developed and is called Enhanced Karnik-Mendel Algorithm (EKM) [96]. Alternatives to these algorithms include the one introduced by Greenfield [97] known as the collapsing method, uncertainty bound method [98], the WuTan (WT) method [99], the Nie-Tan (NT) method [100] and others [101].

The Centroid of an Interval Type-2 Fuzzy Set

Given an IT2 fuzzy set $\tilde{A}$ with MF $\mu_{\tilde{A}}(x)$, its centroid, $C_{\tilde{A}}$, is defined as follows [13]:

$$C_{\tilde{A}} = \int_{\theta_1 \in J_{x_i}} \cdots \int_{\theta_N \in J_{x_N}} 1 \Bigg/ \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i} \tag{2.29}$$

The centroid $C_{\tilde{A}}$ is an interval $C_{\tilde{A}} = [y_l, y_r]$, where y_l is the minimum of all centroids of the embedded T1 FSs in the FOU of $\tilde{A}$ and y_r is the maximum of all centroids of the embedded T1 FSs in the FOU of $\tilde{A}$.

Table 2.1 KM method to find the end points of an IT2 FS [95],[13],[102],[103].

Step	KM algorithm for c_l	KM algorithm for c_r
	$c_l = \min_{\forall \theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]} (\sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i)$	$c_r = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]} (\sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i)$
1	Initialise θ_i by setting $\theta_i = [\underline{\mu}_{\tilde{A}}(x_i) + \overline{\mu}_{\tilde{A}}(x_i)]/2$, $i = 1, 2, \ldots, N$ and then compute $c' = c(\theta_1, \theta_2, \ldots, \theta_N) = \sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i)$	
2	Find $k (1 \leq k \leq N-1)$ such that $x_k \leq c' \leq x_{k+1}$	
3	Compute $c_l(k) = \frac{\sum_{i=1}^{k} x_i \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{k} \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)}$	Compute $c_r(k) = \frac{\sum_{i=1}^{k} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} x_i \overline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{R} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \overline{\mu}_{\tilde{A}}(x_i)}$
4	Check if $c_l(k) = c'$. If yes, stop and set $c_l(k) = c_l$ and $k = L$. If no, go to Step 5.	Check if $c_r(k) = c'$. If yes, stop and set $c_r(k) = c_r$ and $k = R$. If no, go to Step 5.
5	Set $c' = c_l(k)$ and go to Step 2.	Set $c' = c_r(k)$ and go to Step 2.

Note that $x_1 \leq x_2 \cdots \leq x_N$

The centroid $C_{\tilde{A}}$ is an interval T1 FS that is completely characterised by its left and right end points y_l and y_r. KM algorithm [95] shows that y_l and y_r can be found from the LMF and UMF of $\tilde{A}$ as follows:

$$y_l = \frac{\sum_{i=1}^{L} x_i \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{N} x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{L} \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)} \tag{2.30}$$

$$y_r = \frac{\sum_{i=1}^{R} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} x_i \overline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{R} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} \overline{\mu}_{\tilde{A}}(x_i)} \tag{2.31}$$

where $\overline{\mu}_{\tilde{A}}$ and $\underline{\mu}_{\tilde{A}}$ are respectively the upper and lower membership functions of $\tilde{A}$. The L and R are known as switching points that are calculated by the Karnik-Mendel algorithm, are summarised in Table 2.1 [95],[13],[102],[103].

2.6.5 Defuzzifier

After type reduction, defuzzification is usually taken as the mean (average) of the two end points of the type-reduced set, y_l and y_r [27]:

$$y = \frac{y_l + y_r}{2} \tag{2.32}$$

2.7 Transitioning from Type-1 to Type-2 Fuzzy Sets and Systems

2.7.1 Transitioning from Type-1 to Type-2 Fuzzy Sets

FLSs have been commonly attributed with the ability to model and handle uncertainty. Since T2 FLSs have the potential to handle and model 'higher level' of uncertainty than T1 FLSs, they are considered for use in many applications. During the design of T2 FLSs, uncertainties are captured by the FOUs of the T2 FSs. The FOU is a useful concept in IT2 FSs since an IT2 FS is completely described by its FOU [93] and this distinguishes it from a T1 FS.

In terms of the creation of T2 FSs, the common technique is to 'blur' individual T1 FSs to generate a T2 FS. In the literature considering IT2 FLSs, there are many methods used to define an FOU to represent IT2 FSs by transitioning from T1 to IT2 FSs through the technique of blurring. For example, [9],[104] and [77] apply blurring to the T1 MFs while [105] introduces variations to the T1 MFs to generate an IT2 MF. A construction of symmetric or non-symmetric FOUs was proposed in [106–108], where each lower membership function (LMF) and upper membership function (UMF) used was a standard T1 MF such as triangular, trapezoid or Gaussian. IT2 FSs have also been modelled as geometric objects [109],[19] by utilising two poly-lines, one for the UMF of the set and one for the LMF of the set. Ellipsoidal IT2 MFs were created in [110] by introducing an additional parameter into triangular T1 MFs. Diamond shaped MFs [111] were proposed, which have certain values at the endpoints while the middle of the MF captures uncertainty. An FOU of IT2 FS was also created in [112], where the left endpoint and the right endpoint of the LMF and UMF are identical to the T1 FS while the FOU is created using a given parameter of the triangular IT2 FSs. The 'merging' of multiple T1 FSs to create IT2 FSs was also shown in [113]. An FOU of IT2 FS was also adjusted adaptively in [114] based on the amount of uncertainty captured by counting the number of tumor cells.

The transition between T1 and T2 FSs utilising the common technique of blurring has received some criticism. One of the criticisms is that blurring T1 MFs to create IT2 MFs does not always provide improvements in results [74] and, even if there is evidence of success in some applications, this approach fails in others [115].

However, the design of T2 FSs and systems was greatly, and is still, influenced by the advent of the approach of transitioning from T1 to T2 FSs for several reasons, including [113]:

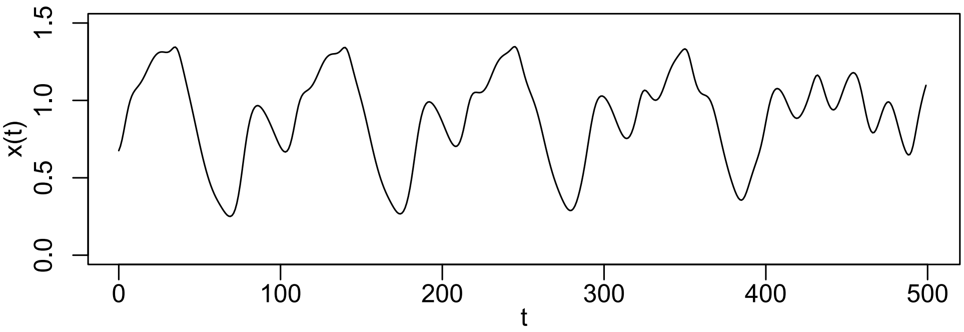
- The system designer is more familiar with T1 than T2 fuzzy systems;
- T1 fuzzy system solutions exist which can be used and transferred to T2 fuzzy systems;
- Generating T1 FSs and systems from data is simple.

In addition, if the transitioning from T1 to T2 FSs is adopted carefully by using 'controlled' blurring for known levels of uncertainty, then such an approach will lead to an FOU that can be related to the uncertainty level and vice versa.

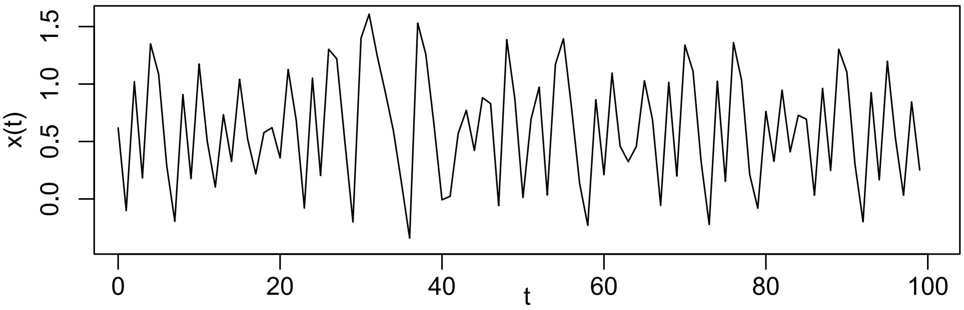
2.7.2 Transitioning from Type-1 to Type-2 Fuzzy Systems

Commonly, there are two different approaches to designing IT2 FLSs [13]: the partially dependent approach and the totally independent approach. The former approach as a method of transitioning from type-1 to type-2 fuzzy systems that starts with the design of a T1 FLS, the parameters of which are then used as the basis for the design of the IT2 FLSs. An advantage of this approach is the potentially faster design of the IT2 FLS, as well as a good comparability between the T1 and IT2 FLSs. The latter approach is used to design IT2 FLS directly without relying on an intermediate T1 FLS and thus a key potential problem with the former approach. This problem is the 'best' IT2 FLS may in fact be very different from the 'best' T1 FLS, i.e. using a T1 FLS as a starting point may in fact be detrimental to the construction of a well-performing IT2 FLS.

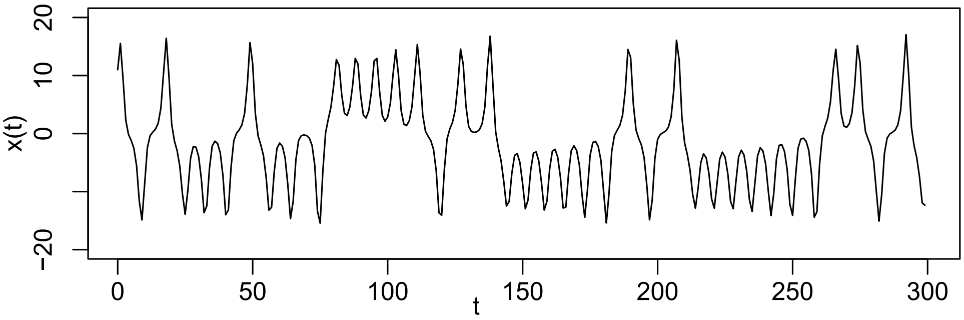
The partially dependent approach has been used to design IT2 FLSs in many applications such as to control a delta parallel robot [104], to control a liquid-level process[116], to predict a time series [117] and others. The totally independent approach has been adopted to enable the T2 FLSs to evolve using an optimisation method.



(a) Mackey-Glass time series



(b) Ikeda time series



(c) Lorenz time series

Figure 2.8 An example of the three time series that will be considered in this thesis (a) Mackey-Glas time series, (b) Ikeda time series and (c) Lorenz time series.

2.8 Time Series Prediction

Time series prediction (TSP) is an important application area that is frequently addressed in the literature [13],[118–120]. It has been applied in many research domains such as weather forecasting, signal processing, economics and production control. It is particularly attractive for the evaluation of algorithms as all underlying parameters (ground truth, noise) can be controlled.

T1 FLSs have been widely used in TSP [121–123]. There are also many works reported in the literature that used T2 FSs to work with time series prediction applications. Kim and Park [124] applied an IT2 FLS to the corrupted Box-Jenkin's gas furnace time series data and compared the results with a T1 FLS. To predict the price of a stock, [125] presented a fuzzy system model approach using T2 FSs compared to T1 FLS. Another use of T2 FLSs was proposed by [126] to improve the prediction of the Taiwan stock index. Karnik and Mendel [9] applied IT2 FLSs to predict the Mackey-Glass (MG) chaotic time series. An investigation into the effect of varying the number of model parameters on the performance of T1 and IT2 FLSs designed to predict the Mackey-Glass time series was proposed by [127] with different levels of additive noise.

Modelling and prediction of chaotic time series such as Mackey-Glass (MG) [128], Ikeda (IK) [129] and Lorenz (LZ) [130] time series are commonly seen in the FLS literature. Figure 2.8 shows samples of those three time series that will be considered in this thesis.

In [131], the authors proposed a designing method for IT2 FLSs based on genetic algorithms and demonstrated their performance using MG chaotic TSP. Jang and Sun [120], showed that FLSs based on 'If-Then' rules can produce a more accurate prediction than the traditional statistical methods and they used the MG time series as a basis for comparison. Gholipour et al. [132] showed how fuzzy neural networks can predict with high accuracy the MG, IK and LZ time series. In [133], the LZ time series was stabilised using an FLS and it was proven that LZ is globally stable by using such system.

2.8.1 Additive Noise

To make the prediction more challenging and more authentic in a real world context, noise can be added to the time series. The level of noise is commonly measured by the Signal-to-Noise Ratio (SNR) where a high SNR refers to a clear signal (low noise) and a low SNR refers to a noisy signal (high amount of noise). As a common and well-defined measure of (one type of) uncertainty, we will use the SNR throughout this thesis. The formula for the SNR (in dBs) [13] is:

$$\text{SNR (in dBs)} = 10 * \log_{10} \left(\frac{\sigma^2{}_{\text{signal}}}{\sigma^2{}_{\text{noise}}}\right), \tag{2.33}$$

where $\sigma^2{}_{\text{signal}}$ is the variance of the signal and $\sigma^2{}_{\text{noise}}$ is the variance of the noise. To find σ_{noise}, we solve (2.33) for σ_{noise} as, i.e.:

$$\sigma_{\text{noise}} = \frac{\sigma_{\text{signal}}}{10^{\left(\frac{\text{SNR}}{20}\right)}} \tag{2.34}$$

Then, the additive noise can be generated from a uniform or Gaussian (normal) distribution as follows:

- Using a uniform random variable with zero-mean in the interval $[-\delta, \delta]$, where $\delta = \sqrt{3}\sigma_{\text{noise}}$. (Note that the variance σ^2 of a uniform random variable in $[-\delta, \delta]$ is $\frac{\delta^2}{3}$) [9].

- Using a Gaussian distribution with zero-mean and standard deviation equal to σ_{noise}.

In this thesis, we generate noisy data using both of these additive noise models.

2.9 Learning and Optimisation of Fuzzy Logic Systems

In this section, we use the terms learning and optimisation of FLSs to refer to the design of an FLS by means of a learning process, generating the rule base, or by an optimisation method, which defines and tunes the FLS parameters.

FLSs have been successfully used in many applications when the systems are too complex to be analysed with mathematical techniques or when the available information is uncertain. In many applications of fuzzy rule-based systems, fuzzy 'if-then' rules and MFs have been obtained and defined from human experts based on the available linguistic and numeric information.

Efforts have been made to automatically generate fuzzy rules in other ways, e.g., using gradient descent methods [134], least-square methods [135],[13], clustering methods [136], neural networks (NNs) [137],[138] and genetic algorithms (GAs) [139–142]. Most of these methods apply long iterative learning or complex rule generation techniques. Wang-Mendel (WM) method [10], proposed an efficient rule creation method with less iterative procedure and adequate performance [143].

Designing and optimising IT2 FLSs parameters and selecting the optimal FOU are common techniques used to improve the performance of systems. However, the process of manually designing and tuning IT2 FLSs is complicated because these systems have more parameters to adjust than their T1 FLSs counterparts, especially with the increase of the MF parameters [140]. Thus, automatic optimisation methods such as those based on neural networks (NNs) [144],[145], genetic algorithms (GAs) [140],[146–149] or simulated annealing (SA) [150–162] have been used in the design of IT2 FLSs in many areas in an effort to increase system performance by finding 'suitable' FOUs.

Since the WM-method and SA are used in this thesis, both the rule creation based on the WM-method and the optimisation of FLSs based on SA will be summarised in the following sections.

2.9.1 Rule Creation for Fuzzy Logic Systems

As mentioned, a core part of designing FLSs beyond the specification of the types and parameters of the FSs employed is the specification of a rule base. In this thesis, we use WM-method [10] to generate a fuzzy rule base from a number of input-output data pairs. The WM-method generates a rule with an associated weight for each training data pair; then, the resulting rule set is truncated by removing the conflicting rules and by using the calculated weights in order to obtain the final rule base.

The WM-method is a general method developed to generate fuzzy rules from numerical data. This method consists of these four steps:

Step 1. Define fuzzy sets to cover the input and output spaces. This is usually done by performing a symmetrical partition of the input/output variable spaces by dividing each universe of discourse into a number of equal partitions. Then, the MF type is chosen and one fuzzy set is assigned to each subspace.

Step 2. Generate one rule from each input-output pair. This forms an initial rule base set that covers each input-output data pair taken from the input-output data set.

Step 3. Assign a degree (weight) to each rule generated in Step 2. Since there are usually lots of data pairs, and each data pair generates one rule, it is highly probable that there will be some conflicting rules, i.e., rules that have the same IF part but a different THEN part. One way to resolve this conflict is to assign a degree to each rule generated from the data pairs, and accept only the rule with the maximum degree from a conflict group.

Step 4. Create the final fuzzy rule base. From the initial rule base, the rule with the maximum weight is chosen for each combination of antecedents forming the final fuzzy rule base.

See [10] and [13] for a more detailed explanation of this approach.

2.9.2 Optimisation of Fuzzy Logic Systems

As discussed, the process of designing an FLS can be optimised with a number of techniques. As a key example of such a technique, we discuss SA, which is used in this thesis.

SA is an optimisation algorithm first introduced in 1953 by Metropolis [163], and then later used by Kirkpatrick et al [163] and Cerny [164], independently, to solve optimisation problems. This algorithm was inspired by the process of metal annealing in which the temperature is controlled to reach an optimal condition. This process usually begins with heating a solid metal to a high temperature, reaching its melting point and then gradually lowering the temperature with a cooling strategy, giving enough time at each cooling stage until the system reaches its minimum energy. Simulated annealing is a global optimisation method that is able to find the global minimum of combinatorial problems [165],[166].

Simulated annealing starts with an initial solution, S_i, selected from the possible search space of solutions. This initial solution, S_i, will be assigned initially as the current solution, S_{curr}, and the best solution, S_{bst}, seen so far, as evaluated by the objective function (fitness function), F, that will return the fitness of the given solutions. Generally, any objective function (fitness function) that can represent how close the FLSs' output to the actual output would be used. One of the most commonly used objective function in this case is an error function. This function is used to find the difference between the output of FLSs and the desired (actual) output by applying a statistical metric such as the root mean square error (RMSE). Then, the smallest fitness of the given solutions that are evaluated by the RMSE function will be taken as the best solution.

A new solution, S_{new}, will then be generated near the current solution (local neighbour) using the neighbour function, N, and finding its fitness $F(S_{new})$. If the fitness of the new solution $F(S_{new})$ is lower than the fitness of the current solution $F(S_{curr})$, then the newly selected solution is accepted and set $S_{curr} = S_{new}$. If the fitness of the new solution is lower than the fitness of the best solution $F(S_{bst})$, then we set $S_{bst} = S_{new}$. Conversely, if the fitness of the new solution is higher than the fitness of the current solution, simulated annealing will accept or reject the new solution based on probability. In this case, simulated annealing uses the Metropolis acceptance procedure [163].

In the Metropolis acceptance procedure, a random number *Rand* in the interval [0,1] will be generated. If the random number is less than the probability of acceptance $P = e^{\frac{-\Delta F}{T}}$, where $\Delta F = F(S_{new}) - F(S_{curr})$ and T is the current temperature, then the new solution will be accepted. Simulated annealing starts with a temperature initialised using a method such as that proposed by [167]. This method uses the probability of acceptance function ($e^{\frac{-\Delta F}{T}}$) to find the initial temperature, T_i. The initial temperature, T_i, can be calculated from the average change in the fitness ΔF result from the number of bad moves M (accepting the worse solution). The average change in the fitness ΔF is given by (2.35):

$$\Delta F = \frac{1}{M} \sum_{i=1}^{M} \Delta F_i \tag{2.35}$$

Then, the initial temperature T_i can be calculated using (2.36) derived from the Metropolis acceptance procedure.

$$T_i = \frac{-\Delta F}{ln(P_0)}, \tag{2.36}$$

where P_0 is the initial probability and chosen to be near 1 (i.e., 0.99) to allow the simulated annealing to start with a high temperature, thereby accepting worse solutions with high probability.

The cooling rate parameter α specifies the rate at which the temperature is reduced using a 'cooling schedule'. At each temperature, a maximum temperature stage length $MaxStg$ is chosen to allow the algorithm to settle into a steady state. The maximum length for the annealing process $MaxAnn$ is the stopping criteria chosen by the user. These are general concepts of simulated annealing; for more details see [168].

2.10 Summary

This chapter introduced the existing literature and the background materials that are used and referred to in the rest of the thesis. The focus of the research and a number of main methods and concepts were presented. First, we introduced the concepts of fuzzy systems and uncertainty. Then, we introduced type-1 fuzzy sets followed by type-1 fuzzy logic systems. Then, the concepts of type-2 fuzzy sets and systems were presented with an overview of the advances in research on their components.

A discussion into the concept of transitioning from type-1 to type-2 fuzzy sets and systems and a review on common techniques of blurring fuzzy sets and designing fuzzy systems were presented.

The time series prediction as an application domain and two different additive noise models were also presented for their use in this thesis. Finally, the main concepts of learning and optimisation of FLSs including the rule base creation method and simulated annealing as an optimisation technique were also presented.

In the next chapter, the investigative approach that covers the main methodology framework by highlighting the main approaches and discussing their main requirements is introduced. Then, the process of design and the evaluation of FLSs based on our proposed approaches is presented.

Chapter 3

Investigative Approach

3.1 Introduction

This chapter presents details on the main investigative framework that covers detailed analysis and presentations of the FOU creation methodologies and their empirical evaluation. It also discusses the potential of a large number of IT2 FS generation algorithms and outlines the subset of three specific cases addressed in this thesis. Thus, this chapter is pivotal for the argument of the whole thesis.

3.2 Requirements for Transitioning from Type-1 to Type-2 Fuzzy Sets and Systems

As discussed in Chapter 2 (Section 2.7), there are many methods used to define an FOU to represent IT2 FSs by transitioning from T1 to IT2 FSs through the common technique of blurring. In this thesis, we adopt design methodologies for controlled

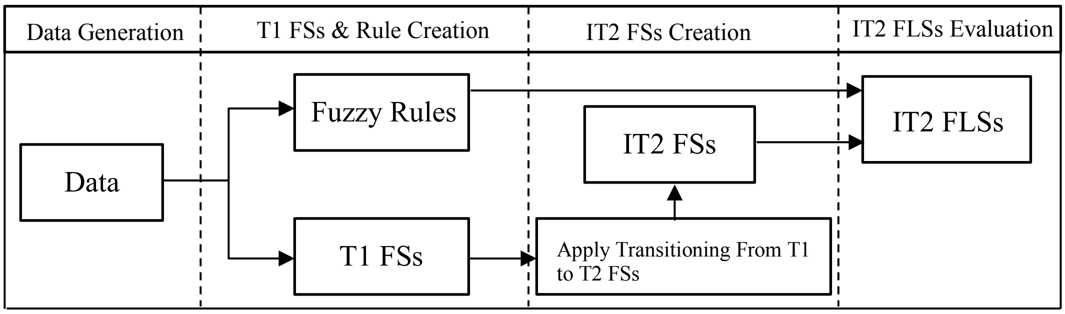


Figure 3.1 Illustration of the generic process of designing and evaluating IT2 FSs and systems

blurring of T1 MFs, i.e., for driving IT2 MFs where the size of the primary membership interval remains at a constant size over the support of the LMF.

In this thesis, we adopt the partially dependent approach detailed in Chapter 2 (Section 2.7.2) since the preservation of the basic structure (number of MFs, rule base) is essential to our aim of comparing a series of FLSs that range from T1 to IT2 FLSs with increasing FOUs. We acknowledge the risk that the best performance of the IT2 FLSs may be achieved without relying on a T1 implementation, but we feel that for the proposed investigation (which is not about finding the most performance), the use of the partially dependent approach is warranted and suitable.

The general process that is used to implement the transitioning from T1 to IT2 FLSs is illustrated in Figure 3.1. Detail on the individual aspects of each stage shown in the figure will be further detailed in the next section. In this section, we will specifically outline the requirements of transitioning from T1 FSs to IT2 FSs shown in the third stage of the process.

Following a partially dependent approach, the main issue after a T1 system has been designed as shown in Figure 3.1, is the specific requirements of the transitioning from the T1 MFs to IT2 MFs, in other words, the requirements of IT2 FSs to capture and model the uncertainty. While these requirements are application dependent, a key set of commonly applicable requirements are the following:

1. To adequately reflect the uncertainty affecting a given variable in the resulting IT2 FSs.

2. To maintain a given level of uncertainty in the FOU of an IT2 FS throughout its support as far as possible. This requirement is designed to capture that if an uncertainty level is known in relation to a specific FS, then within the resulting IT2 FS, this uncertainty should be reflected at this level.

3. To preserve the shape of the original T2 FS. In many applications, it may be desirable to preserve the original shape of the T1 FSs as closely as possible enabling an intuitive comparison between the T1 and T2 FSs. For example, if the original T1 FSs are triangular, then the resulting IT2 FSs should also be triangular.

4. To enable the optimisation of the resulting IT2 FSs in response to changing levels of uncertainty. This final requirement is particularly relevant in applications where the uncertainty levels vary or are poorly known.

As was noted above, the actual selection of which requirements are essential is application dependent (e.g., requirement 3 may not be relevant in a control application). Thus, in this thesis, each of these requirements is addressed in specific Chapters, each of which is introduced briefly below.

3.2.1 The Uncertainty Indicator

In order to address requirement 1, we initially establish the concept of the 'Uncertainty Indicator' with the view to capturing the level of uncertainty reflected in a given IT2 FS.

A number of common types of MFs for FSs exist in the literature as discussed in Chapter 2 (Section 2.3.1). Two versions of such MFs are commonly used when considering IT2 FSs: MFs with an uncertain mean (centre for the triangular case) and those with uncertain standard deviation (spread for the triangular case). In this thesis, the triangular, trapezoidal and Gaussian MFs are considered. Specifically, we consider the triangular and Gaussian MFs in Chapter 4 and Chapter 6 and the triangular, trapezoidal and Gaussian MFs in Chapter 5. So, we have added the trapezoidal MFs in the investigation in Chapter 5 to study its behaviour with two different approach as will be detailed in Chapter 5.

In order to study the uncertainty modelling of IT2 FSs, we introduce the concept of the Uncertainty Indicator $U(x)$. $U(x)$ is principally the size of the primary membership interval at a given x for a given IT2 FS, i.e.:

$$U_{\tilde{A}}(x) = |J_x| \tag{3.1}$$

where $|J_x| = \overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)$.

The behaviour of the Uncertainty Indicator $U(x)$ on standard triangular and Gaussian IT2 MFs is illustrated in Figure 3.2 and Figure 3.3, respectively. For the triangular case, Figure 3.2 shows IT2 FSs with uncertain spread and uncertain centre respectively. For the Gaussian case, Figure 3.3 shows IT2 FSs, also with uncertain spread and uncertain centre, respectively. In both cases, it can be seen that the $U(x)$ value is not constant over the support of the LMFs ($\underline{\mu}_{\tilde{A}}(x) > 0$) of the given sets. This highlights that the FOU is not constant, especially in this 'central' region of the FSs.

We ascertain that, in cases where we know a specific and constant level of uncertainty associated with a given variable (e.g., level of measurement imprecision) modelled by an (IT2) FS, then, this FS should reflect the present level of uncertainty throughout its membership as far as possible (i.e., excluding its edges). Thus, an appropriate FOU construction methods should give rise to an equal amount of uncertainty to be captured over the primary domain of the MF at least within this 'core' of the FS, i.e., the support of the LMF. This directly reflects requirement 2 in Section 3.2.

To achieve this, we propose two design methodologies which systematically vary (blur) the size of the FOU of IT2 FSs, enabling the systematic relating of FOU size

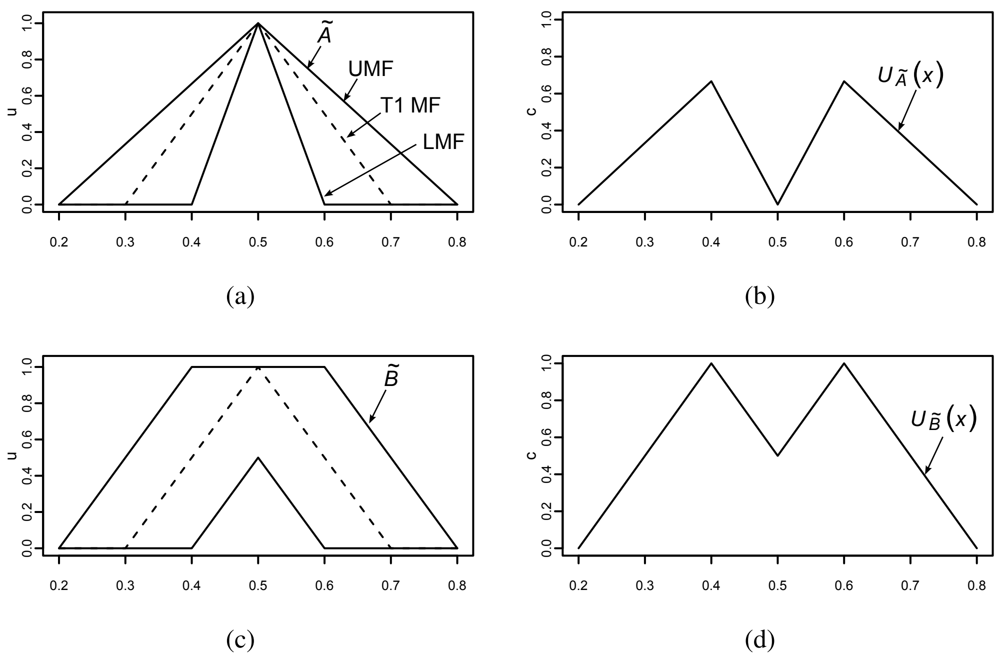


Figure 3.2 Triangular IT2 FSs and their Uncertainty Indicators. (a) IT2 FS $\tilde{A}$ with uncertain spread and (b) its Uncertainty Indicator $U_{\tilde{A}}(x)$, (c) IT2 FS $\tilde{B}$ with uncertain centre and spread and (d) its Uncertainty Indicator $U_{\tilde{B}}(x)$. The dotted lines in both (a) and (c) are the principal T1 FSs.

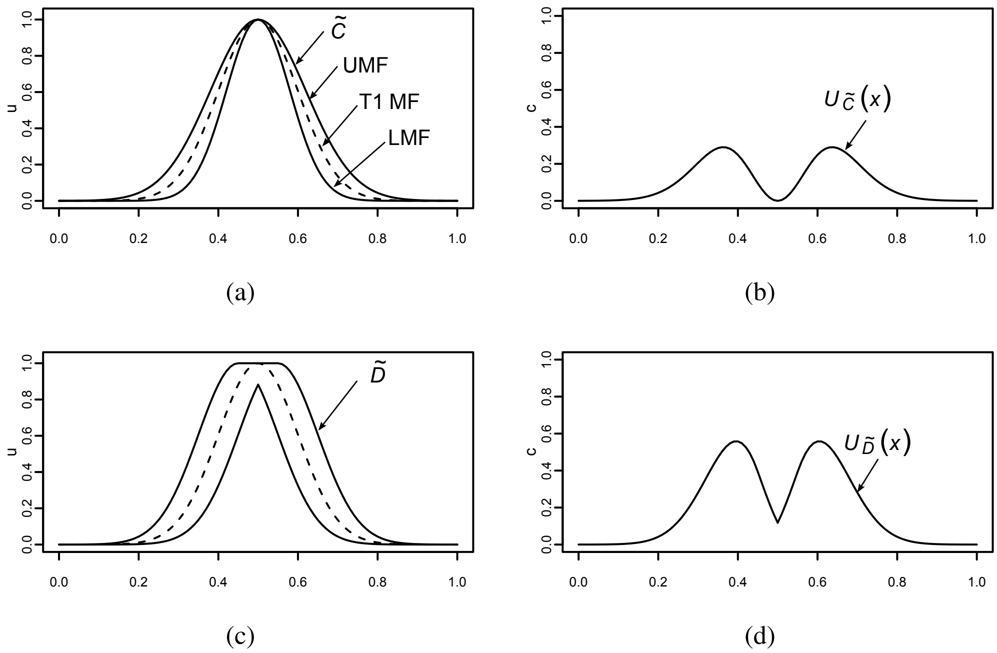


Figure 3.3 Gaussian IT2 FSs and their Uncertainty Indicators. (a) IT2 FS $\tilde{C}$ with uncertain standard deviation and (b) its Uncertainty Indicator $U_{\tilde{C}}(x)$, (c) IT2 FS $\tilde{D}$ with uncertain mean and (d) its Uncertainty Indicator $U_{\tilde{D}}(x)$. The dotted lines in both (a) and (c) are the principal T1 FSs.

to levels of uncertainty and vice versa. The first approach is the controlled blurring method used to create an FOU of IT2 FS from an original T1 FS and maintain the level of uncertainty captured in the primary memberships of the FS constant but not maintain the original T1 MF shape (i.e., requirements 1 and 2 only), while, the second approach which is the shape-preserving controlled blurring method that follows a similar methodology as the first approach and also keeping the IT2 FS shape (LMF and UMF) the same as the original T1 MF (i.e., requirements 1, 2 and 3). These two features of the latter approach are important as they enable the systematic relating of FOU size to levels of uncertainty and vice versa and enable an intuitive comparison between the T1 and T2 FSs.

The outline of the first approach that satisfies both requirements, 1 and 2 above is introduced next.

3.2.2 The Controlled Blurring Method

In an effort to obtain an IT2 FS with a constant size of FOU over the 'core' of the FS, we propose an FOU construction method based on a fixed uncertainty level parameter $c \in [0,1]$ used to create an FOU of a given size around a (principal) T1 MF. In other words, we are proposing an approach for the controlled blurring of type-1 MFs. The proposed approach enables the transition from T1 to IT2 FLSs by varying the size of the FOU of their respective FSs, while keeping the size of the FOU over the FS as constant as possible. This is an important feature of this method as it enables the relating of FOU size to levels of uncertainty and vice versa. We will refer to this approach as the controlled blurring (CB) method throughout the thesis.

Figure 3.4, shows an example of how the transition from T1 to IT2 FSs is achieved by applying controlled blurring to T1 MFs. In order to create the IT2 FSs based on the uncertainty parameter c and a T1 MF, we employ certain requirements to ensure that the resulting FOU is uniform (constant) within the 'core' (i.e. the support of the LMF) of the IT2 FSs. These requirements include: the value of the Uncertainty Indicator, $U(x)$, should be constant over the support of the LMFs, i.e., satisfying the uncertainty indicator requirement, and the values of both LMFs and UMFs are bounded in the interval [0,1] for the given values of c.

From the T1 FS shown in Figure 3.4, two IT2 FSs are created by including the pre-specified FOU size parameter $c = 0.2$ and $c = 0.8$. Note that $c = 0$ results in a T1 FS with the original MF (i.e., uncertainty indicator equal 0), while $c = 0.2$ and $c = 0.8$

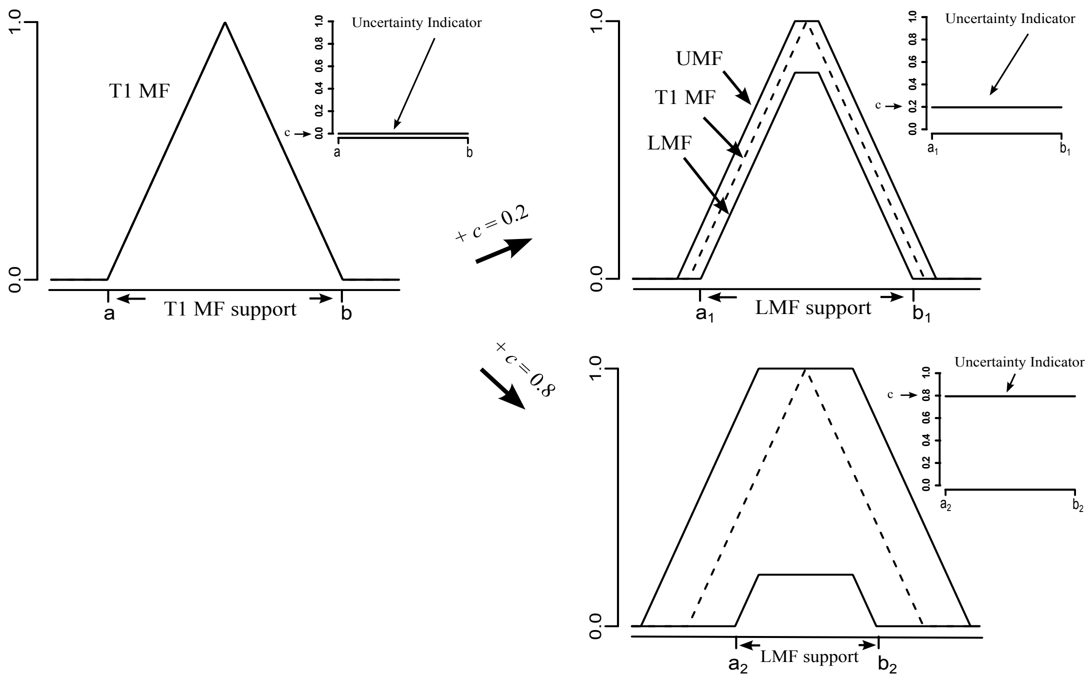


Figure 3.4 Example of the controlled blurring of a T1 MF to create IT2 FSs using the CB method with two different c values. The uncertainty indicator domain is the LMF support and its value for each FS is shown (i.e., identical to c value).

result in IT2 sets with uncertainty indicator values equal to the given c (i.e., $c = 0.2$ and $c = 0.8$) within the support of the LMFs of the IT2 FSs. This illustrates that the FOU in this region of the FSs is uniform (constant) as designed, unlike in the standard blurring (SB) approach as shown in Figure 3.2.

The general scope of this method is discussed above. The functions that are used to create IT2 FSs from an initial T1 FS and the process of obtaining the UMF and LMF will be detailed in Chapter 4.

Having illustrated the general features of the CB method, it is important to notice that the shape of the IT2 FSs resulting from this method is changed. In Figure 3.4, we see that the resulting UMF and LMF of the IT2 FS is converted to trapezoidal, whereas the original T1 MF is triangular. Keeping the original MF shape would enable a more intuitive comparison between the T1 and T2 FSs. This observation leads us to propose the next method that enables shape-preserving while applying the controlled blurring technique. The outline of the second approach that satisfies requirements, 1, 2 and 3 above is introduced next.

3.2.3 The Shape-Preserving Controlled Blurring Method

In an effort to find a controlled blurring method, addressing the shortcoming of CB method outlined above and thus satisfying requirement 3, a new shape-preserving controlled blurring method is proposed. In order to create the IT2 FSs based on a bounded value of uncertainty δ, a fixed parameter c and a T1 MF, we employ certain requirements to ensure that the resulting FOU is uniform (constant) within the 'core' (i.e. the support of the LMF) of the IT2 FSs, and that the UMF and LMF maintain the original T1 MF shape. This will enable both the relating of FOU size to levels of uncertainty and an intuitive comparison between the T1 and IT2 FSs. These requirements include: the value of the Uncertainty Indicator, $U(x)$, should be constant over the support of the LMFs (i.e., satisfying uncertainty indicator requirement) and the resulting UMF and LMF of the IT2 FS should keep the original T1 MF shape (i.e., satisfying the shape-preserving requirement). This will enable the creation of an FOU of a given size around a (principal) T1 MF and will maintain the same shape of T1 MF for both the UMF and LMF as the FOU size increases. We will refer to this approach as the shape-preserving controlled blurring (SP-CB) method throughout the thesis.

A general illustration of creating IT2 FSs using the SP-CB is depicted in Figure 3.5. We start the design of the IT2 FSs using SP-CB method by including the FOU size parameter c and the bounded value of uncertainty δ to form the IT2 FSs of the system. Note that when both c and δ are equal to zero, then the result is the original T1 FS

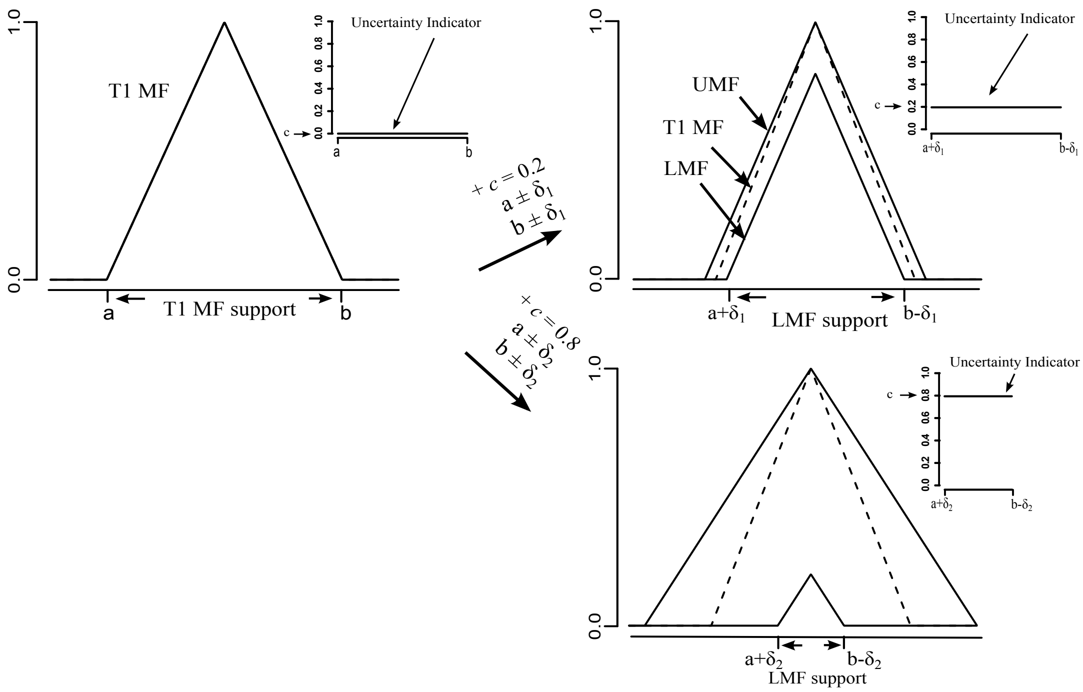


Figure 3.5 Example of the controlled blurring of T1 MF to create IT2 FSs using the SP-CB method by employing two different c and δ values. The uncertainty indicator domain is the LMF support and its value for each FS is shown (i.e., identical to c value).

MF, while $c = 0.2$ and $c = 0.8$ result in IT2 FS with uncertainty indicator values equal to the given c (i.e., $c = 0.2$ and $c = 0.8$) within the the support of the LMFs of the IT2 FSs. In order to preserve the original shape of the membership function (T1 MF) as we add a certain value of the FOU size parameter c to it to create an IT2 FSs, we construct the UMF and the LMF of the IT2 FSs by adding/subtracting a specific value of (δ) from the T1 MF parameters. The δ parameter is used to define the endpoints in the fuzzy MFs (see Figure 3.5). Its value can be obtained experimentally, so that, for a given value of c, the uncertainty indicator, $U(x)$, is constant (equal to c) over the support of the LMF.

The general features and requirements of this method are discussed above and the functions that are used to create IT2 FSs from an initial T1 FS and obtain the UMF and LMF will be detailed in Chapter 5.

If we have established an appropriate FOU in the existing system and the environment or conditions change, in this case, the FOU refinement is crucial to enable better adoption in the new conditions. Thus, an optimised controlled blurring method used for optimising IT2 FLS, in which the controlled blurring (CB) method is adopted, and the design parameters are tuned through an optimisation method is introduced.

The outline of the third approach that satisfies both requirements, (1) and (4) above is introduced next.

3.2.4 The Optimised Controlled Blurring Method

In this section, a method for designing IT2 FLS, in which the CB method (presented in the previous section (3.2.2)) is adopted, and then the design parameters are tuned through an optimisation method (e.g., simulated annealing). This is done to refine the FSs for existing systems and to select the optimal FOU size in an application with varying levels of uncertainty, enabling better adaptation in the real world and thus addressing requirement 4 in section 3.2. The CB approach has few parameters to be tuned as only a single extra parameter (i.e., FOU size parameter c) is used to define the IT2 MFs.

One of the challenges in modelling a T2 FLS is the problem of defining the MF parameters and their FOUs given noisy data or imperfect measurements. This challenge is increased by the complexity which arises from the increase in the number

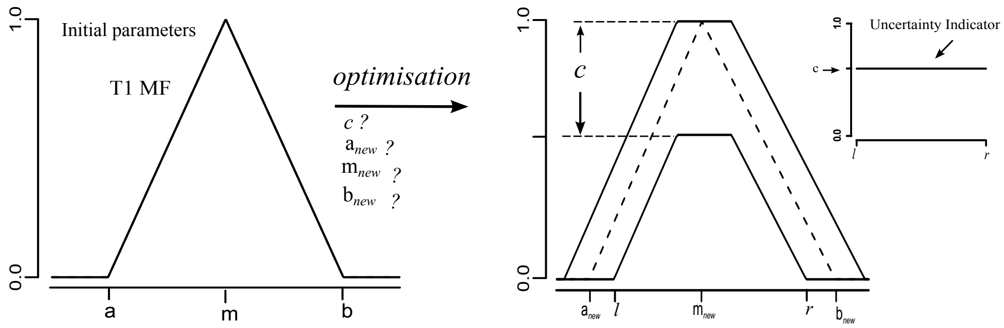


Figure 3.6 Example of the creation, optimisation of IT2 FSs and selecting the FOU size using the O-CB method based on initial parameters given by a T1 FS. The uncertainty indicator domain is the LMF support and its value for the FS is shown (i.e., identical to c value).

of parameters of IT2 MFs to be tuned. Also, the process of manually designing and tuning IT2 FLSs is complicated.

To simplify the complexity which arises from the increase in the number of parameters to be tuned in IT2 MFs, the CB method presented in the previous section (3.2.2) is adopted here. This method is designed to create IT2 FSs with a uniform FOU over the support of the lower MF (LMF) 'core' of the fuzzy set by incorporating a fixed FOU size parameter $c \in [0,1]$ to create an FOU of a given size around a T1 MF. This method has fewer parameters than usual IT2 FSs. It is dependent partially on T1 FSs (i.e., usual T1 MF parameters), introducing only one extra parameter (FOU size parameter c). The implementation of the CB method utilising an automatic optimisation method will increase the performance of the FLSs by finding 'suitable' FOUs and better adaptation in real world applications. We will refer to this approach as the optimised controlled blurring (O-CB) method throughout the thesis.

A general illustration of the O-CB method is depicted in Figure 3.6. From the T1 FS shown in Figure 3.6 with parameters used as an initial solution, an IT2 FS is created by including the FOU size parameter c and new parameters generated near the initial T1 MF parameters, tuned and selected by an optimisation method. This is the general scope of this method and it will be detailed in Chapter 6.

Having considered the transition process from T1 to T2 FSs and its requirements by presenting the three approaches, CB, SP-CB and O-CB, in the next section, we proceed to the FLS design and evaluation process that will be applied to each one of the proposed methods discussed. This process provides significant insight into the relationship levels/amount of uncertainty and appropriate FOU size in a given domain.

In order to examine the behaviour of the proposed techniques (design/optimising), a general systematic design and evaluation methodology is detailed below. The designs of T1/IT2 FLSs considered with respect to different noise levels and thus different desired FOU sizes are presented. Details on the individual aspects of each approach are given in the next chapters.

3.3 Fuzzy Logic System Design and Evaluation

This section details the systematic processes carefully designed to enable the three major components of the research of this thesis: CB, SP-CB and O-CB methods.

The following systematic processes describe the initial design of the T1 FLS for a given application and its subsequent transformation to one or more IT2 FLSs by applying the IT2 FSs generation/optimisation methods: CB, SP-CB and O-CB. Specifically, the design of multiple T1/IT2 FLSs by creating different size of FOUs with respect to different noise levels are presented, as well as the design and optimisation of IT2 FLS parameters and selecting the optimal FOU based on one of the FOU creation methods (e.g., CB) that has less parameters to be tuned are presented.

The general process used to implement and evaluate these design methods as illustrated in Figure 3.1 can be summarised in the four subsections (3.3.1 - 3.3.4) detailed below.

3.3.1 Data Generation

Training and testing data sets are generated from the system under study (i.e., time series). Usually, there are two different cases that are considered when preparing the training and testing data. The first approach is used to generate the training data and is kept noise free, while testing data is subsequently injected with different noise levels. The second approach is used to generate both the training and testing data and both are injected with different noise levels. Training and testing using the same noise level (i.e. SNR) is the case in most real world applications, but it is also possible that a system trained using noise-free (NF) data, for example as in laboratory conditions, can then be used in real world conditions (noisy).

The training data is used to train the system (generating rules, initialising the FLS parameters and training the system during the optimisation process) under two conditions: NF data ('ideal conditions') and noisy data (as expected in real world situations) at different noise levels. The testing data sets (noise free and corrupted by noise) are used to test and evaluate the performance of a designed system in the face of a given level of noise (uncertainty).

3.3.2 Type-1 Fuzzy Sets Design and Rule Base Creation

First, T1 FSs are created, either by an expert (as in our case as detailed in Chapters 4, 5 and 6) or through an automatic method (e.g., a genetic algorithm). The initial T1

FSs are created using the training data maximum and minimum values to divide each input space into the chosen number of fuzzy sets, providing sufficient overlap between them. Second, the training data is used to generate the rules, using for example, the Wang-Mendel (WM) method [10]. The created rule base is then used for all subsequent FLSs.

3.3.3 Interval Type-2 Fuzzy Sets Creation

The T1 FSs are extended into IT2 FSs using the existing T1 MFs as a basis following the partially dependent design introduced in Chapter 2 (Section 2.7.2). By applying the transitioning from T1 to T2 FSs requirements for each FOU creation method detailed earlier in this chapter, the results are IT2 FSs with a uniform (constant) FOU over the 'core' of their FSs, enabling a constant amount of uncertainty to be captured over its primary domain.

3.3.4 Fuzzy Logic Systems Evaluation

After finishing the creation/optimising of each of the IT2 FLSs with the chosen FOU size parameter, we evaluate their performances using the pre-generated testing data for different uncertainty/noise levels.

At each noise level, the performance evaluation is repeated a number of times to account for the random character of the noise injection. The performance of the generated design(s) is evaluated, for example using the Root Mean Square Error (RMSE) as a measure of performance that has been applied in many variety of experimental scenarios and across a wide range of simulated and real world applications. The RMSE is given in (3.2) as:

$$RMSE = \sqrt{\frac{1}{Q}\sum_{i=1}^{Q}\left[Y_i - \hat{Y}_i\right]^2} \tag{3.2}$$

where Q is the total number of data points (testing data), Y_i is the actual output (from the testing data) and $\hat{Y}_i$ is the crisp output of the FLS.

The average of the RMSEs is then calculated over all iterations for each FLS at each uncertainty/noise level in order to enable the best mapping between the FLSs (with different FOU sizes) and the noise levels.

3.4 Summary

In this chapter, the transitioning from type-1 to type-2 fuzzy sets and systems requirements were presented, relating to objectives 1, 2 and 4 from Chapter 1. Discussions related to these requirements that may be applicable in order to enable the uncertainty modelling of IT2 FSs in specific applications were detailed. Three specific IT2 FSs creations methods, namely, the controlled blurring, shape-preserving controlled blurring and optimised controlled blurring, satisfying these requirements were briefly outlined. Detail on the individual aspects of each approach outlined will be further investigated in the next chapters of the thesis.

The systematic process to design/optimise and evaluate IT2 FLSs has been established, creating a general framework that will be employed in the subsequent chapters.

In the following chapter, we conduct a series of in–depth experiments using the described systematic IT2 FLSs design and evaluation process outlined in Section (3.3) specifically, by detailing and implementing the controlled blurring (CB) method highlighted in Section (3.2.2) in the context of time series prediction.

Chapter 4

An Investigation into the Controlled Blurring for the Generation of Interval Type-2 Fuzzy Sets

4.1 Introduction

This chapter investigates the relationship between FOU size and uncertainty levels by exploring the first approach, the controlled blurring (CB) method that was initially presented in Chapter 3 (Section 3.2.2) and satisfies requirements 1 and 2 in Section 3.2. The CB method is used for designing IT2 FSs so that their FOUs can capture varying levels of uncertainty/noise.

To fulfil the first objective declared in Chapter 1, we provide insight into the challenging question through a detailed investigation into the ability of both types of FLSs to capture and model different levels of uncertainty/noise through varying the size of the FOU of the underlying fuzzy sets from T1 FSs to very ‘wide’ IT2

FSs. In order to explore the relationship between FOU size and various levels of uncertainty/noise, we conduct a series of experiments forecasting the Mackey-Glass (MG) [128], Ikeda (IK) [129] and Lorenz (LZ) [130] time series.

In the next section, the controlled blurring method is detailed. In section 4.3, the experimental data that is used to conduct the experiments is presented. A series of experiments that illustrate the FOU construction approach and follow the IT2 FS design and evaluation process (highlighted in Chapter 3 (Section 3.3)) will be conducted in section 4.4, and the results will also be included. The results will then be analysed and discussed in section 4.5, followed by the chapter summary in section 4.6.

4.2 The Controlled Blurring Method Details

In an effort to obtain an IT2 FS with a uniform (constant) FOU over the ‘core’ of the FS, we propose a controlled blurring (CB) method that is used to construct an FOU of employed IT2 FS based on a fixed uncertainty level parameter $c \in [0,1]$ that is used to create an FOU of a given size around a (principal) T1 MF, thus satisfying requirements 1 and 2 in Section 3.2. In other words, we are proposing an approach for the controlled blurring of type-1 MFs. The proposed approach enables the transition from T1 to IT2 FLSs through varying the size of the FOU of their respective FSs while keeping the size of the FOU over the FS as constant as possible. This is an important feature of this method as it enables the relating of FOU size to levels of uncertainty and vice versa.

We start the design of the IT2 FSs by including the FOU size parameter c to form the IT2 FSs of the system. Note that $c = 0$ results in a T1 FS with the original MF while $c = 1$ results in an IT2 set with a very wide FOU (i.e. the FOU covers the entire primary membership).

In order to create the IT2 FSs based on the uncertainty parameter c and a commonly used T1 MF (e.g., triangular or Gaussian), we employ (4.1) and (4.2) shown below to create the resulting UMF and LMF respectively. Note that the minimum operation in (4.1) and the maximum operation in (4.2) ensure that the values of both LMF and UMF are bound by the interval [0,1]. Also, in (4.2), note that the minimum operation prevents the LMF from exceeding the value of $1.0-c$ (its maximum value). Both (4.1) and (4.2) have been designed to ensure that the resulting FOU is uniform (constant) within the 'core' (i.e. the support of the LMF) of the IT2 FSs. The UMF, $\overline{\mu}_{\tilde{A}}(x)$ and the LMF, $\underline{\mu}_{\tilde{A}}(x)$ of IT2 FS $\tilde{A}$ can be obtained as follows:

$$\overline{\mu}_{\tilde{A}}(x) = \min\left(\mu_A'(x) + \frac{c}{2}, 1.0\right) \tag{4.1}$$

$$\underline{\mu}_{\tilde{A}}(x) = \min\left(\max\left(\mu_A'(x) - \frac{c}{2}, 0\right), 1.0-c\right), \tag{4.2}$$

where $\mu_A'(x)$ relates to the T1 MFs shown in (4.6), (4.7) and (4.8) and c is the FOU size parameter.

An initial T1 triangular MF A specified by three parameters $\{a, m, b\}$ can be defined as:

$$\mu_A(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m-a} & a \leq x \leq m \\ \frac{b-x}{b-m} & m \leq x \leq b \\ 0 & x \geq b \end{cases}, \tag{4.3}$$

where, a is the left endpoint, m the centre and b the right endpoint.

An initial T1 trapezoidal MF A specified by four parameters $\{a, m_1, m_2, b\}$ can be defined as:

$$\mu_A(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m_1-a} & a \leq x \leq m_1 \\ 1.0 & m_1 \leq x \leq m_2 \\ \frac{b-x}{b-m_2} & m_2 \leq x \leq b \\ 0 & x \geq b \end{cases}, \tag{4.4}$$

where a is the left endpoint, m_1 the left centre, m_2 the right centre and b the right endpoint.

An initial T1 Gaussian MF A specified by two parameters $\{\sigma, m\}$ can be defined as:

$$\mu_A(x) = e^{\frac{-(x-m)^2}{2\sigma^2}}, \tag{4.5}$$

where, σ is the standard deviation and m the mean.

In order to use (4.3), (4.4) and (4.5) as the initial T1 MFs for the resulting IT2 FSs in conjunction with (4.1) and (4.2), we first extend the initial T1 MFs to avoid UMFs which do not tend to 0 on either side of the centre/mean as shown in Figure 4.1(a) and 4.1(c). Figure 4.1 illustrates the method of extending T1 MFs (triangular and Gaussian) to have negative 'membership values' (Figure 4.1(b) and Figure 4.1(d)), based on (4.6) and (4.8) respectively. Employing the resulting 'extended' MFs makes it straightforward to use (4.1) and (4.2) in conjunction with the FOU parameter c to generate the actual IT2 sets with a legible shape.

Note that for Gaussian MFs, the new principal (T1 MF) dotted line in Figure 4.1(d) follows the initial (original) Gaussian principal (T1 MF) closely after (4.8) has been applied - but it now enables the straightforward generation of the IT2 FS's UMF and LMF using (4.1) and (4.2).

$$\mu'_A(x) = \begin{cases} \frac{x-a}{m-a} & a \leq x \leq m \\ \frac{b-x}{b-m} & m \leq x \leq b \end{cases} \tag{4.6}$$

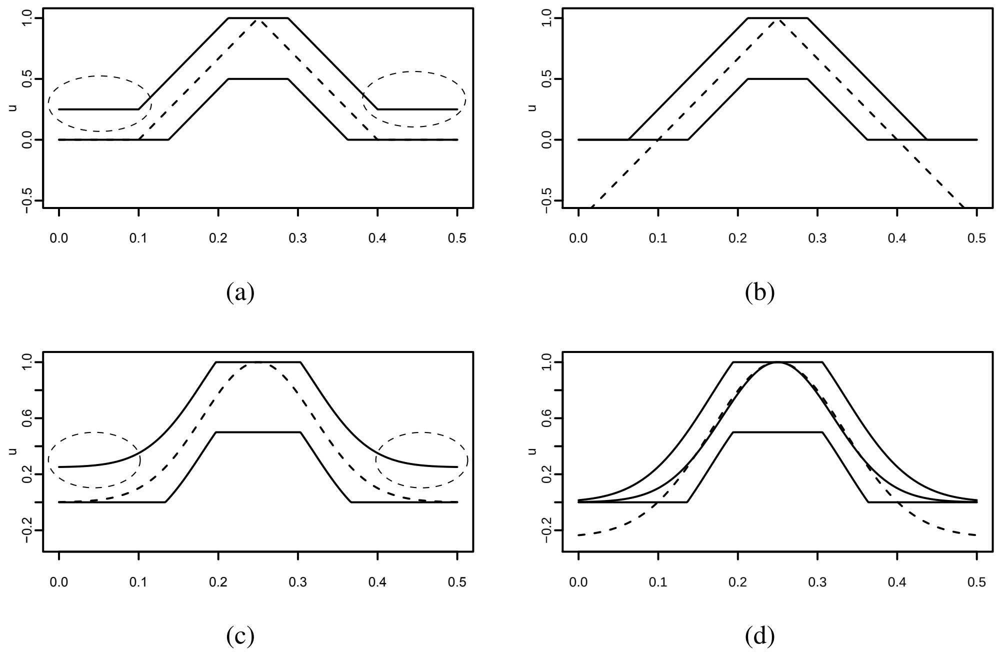


Figure 4.1 An illustration of the IT2 FS design for a T1 MF (dotted line) and parameter $c = 0.50$. (a) IT2 triangular MF designed using (4.1) and (4.2), using IT2 MF based on (4.3). (b) IT2 triangular MF designed using (4.1) and (4.2), using IT2 MF based on adopted version of (4.3) i.e., (4.6). (c) IT2 Gaussian MFs designed using (4.1) and (4.2), using IT2 MF based on (4.5). (d) IT2 Gaussian MFs designed using (4.1) and (4.2), using IT2 MF based on adopted version of (4.5) i.e., (4.8). The dotted circles around the end of the UMFs in (a) and (c) show the potential problem of the UMFs not tending to 0 when using (4.3) and (4.5) directly as initial T1 MF. So, the derived IT2 MFs in (b) and (d) based on the extended T1 MFs are used in our approach (i.e., CB method).

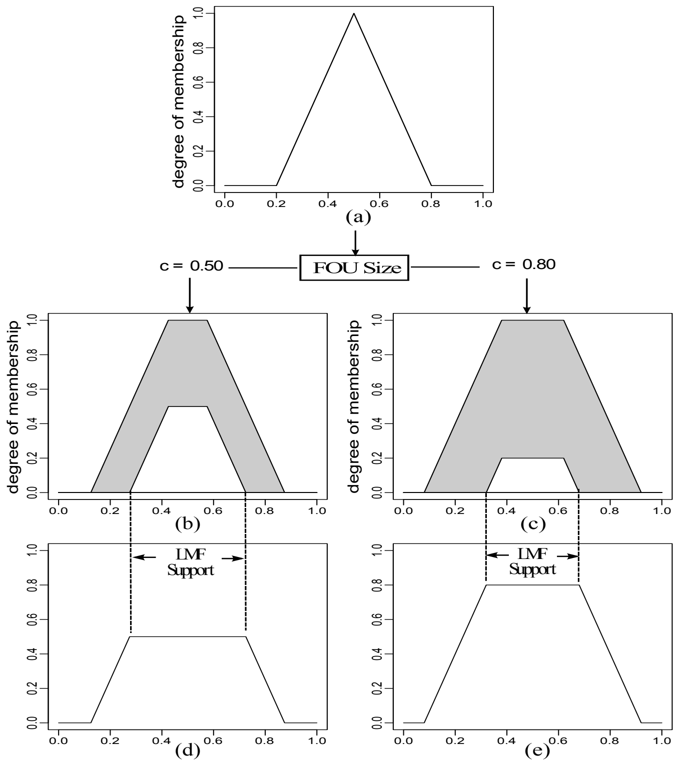


Figure 4.2 An illustration of the uniform design of triangular IT2 MFs using CB method. (a) Initial T1 FS, (b) IT2 obtained using FOU size parameter $c = 0.5$, (c) IT2 FS for $c = 0.8$, (d) Uncertainty Indicator obtained from IT2 FS with FOU size parameter $c = 0.5$ and (e) Uncertainty Indicator for $c = 0.8$.

$$\mu'_A(x) = \begin{cases} \frac{x-a}{m_1-a} & a \leq x \leq m_1 \\ 1.0 & m_1 \leq x \leq m_2 \\ \frac{b-x}{b-m_2} & m_2 \leq x \leq b \end{cases} \tag{4.7}$$

$$\mu'_A(x) = (1+\frac{c}{2}) * (e^{\frac{-(x-m)^2}{2*((1+\frac{c}{3})*\sigma)^2}}) - \frac{c}{2}, \tag{4.8}$$

where c is the FOU size parameter.

Having illustrated the functions used to create IT2 FSs from the initial T1 FSs, Figure 4.2 and Figure 4.3 provide illustrations of the resulting IT2 MFs for the triangular and Gaussian cases respectively. Both are designed with the FOU parameter

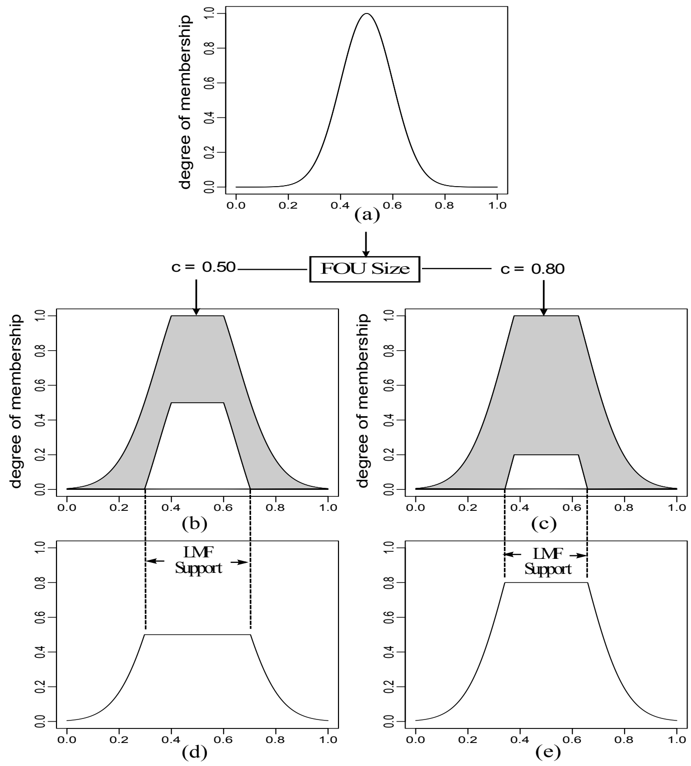


Figure 4.3 An illustration of the uniform design of Gaussian IT2 MFs using CB method. (a) Initial T1 FS, (b) IT2 obtained using FOU size parameter $c = 0.5$, (c) IT2 FS for $c = 0.8$, (d) Uncertainty Indicator obtained from IT2 FS with FOU size parameter $c = 0.5$ and (e) Uncertainty Indicator for $c = 0.8$.

$c = 0.5$ and $c = 0.8$ using equations (4.1) and (4.2). From the T1 FS shown in Figure 4.2(a) and Figure 4.3(a), two IT2 FSs are created by including the pre-specified FOU size parameter $c = 0.5$ in Figure 4.2(b) and Figure 4.3(b) and FOU size parameter $c = 0.8$ in Figure 4.2(c) and Figure 4.3(c). The UMF and LMF are obtained using equations (4.1) and (4.2).

By using equation (3.1), the Uncertainty Indicator for all the IT2 FSs is obtained and depicted in Figure 4.2(d), Figure 4.2(e), Figure 4.3(d) and Figure 4.3(e). In all cases, it can be seen that the Uncertainty Indicator, and thus the size of the primary membership interval, is now constant over the support of the LMFs at a value equal to c (note that, in Figure 4.2(d) and Figure 4.3(d), $c = 0.5$ and in Figure 4.2(e) and Figure 4.3(e) $c = 0.8$).

4.3 Experimental Data

In the following, the three time-series that are used to provide a platform to explore the behaviour of FLSs with different FOU sizes in respect to different levels of uncertainty/noise are presented.

4.3.1 Mackey-Glass Time Series

The Mackey-Glass (MG) time series is a chaotic time series proposed in [128] containing a first-order differential-delay equation to model a physiological systems (Equation (4b) in [128]). It is obtained from the non-linear equation:

$$\frac{dx(t)}{dt} = \frac{a * x(t-\tau)}{1 + x^n(t-\tau)} - b * x(t), \tag{4.9}$$

where a, b and n are constant real numbers, t is the current time and τ is the delay time. For $\tau \leq 17$, the system is known to exhibit deterministic/periodic behaviour which turns chaotic when $\tau > 17$. To obtain simulation data, (4.9) is used in this study with the following parameters: $a = 0.2$, $b = 0.1$, $\tau = 30$ and $n = 10$. It is solved using Euler's method [169] with a step size equal to 1.0 and where the initial values of $x(t)$ for all values of $t \leq \tau$ are set to 0.9. The chosen setting for MG parameters is the same as [9],[13].

The MG time series for $\tau > 17$ is used as one of the benchmark problems for time-series prediction in fuzzy logic [9],[13],[120],[170] and neural network [118],[119],[171] areas.

4.3.2 Ikeda Time Series

The original Ikeda (IK) map was proposed in [129] as a model used to simulate light travelling around a ring cavity (non-linear system). The IK time series can be generated by the following 2-dimensional real form equations:

$$\dot{x} = 1 + \mu * (x * \cos(t) - y * \sin(t)), \tag{4.10a}$$

$$\dot{y} = \mu * (x * \sin(t) + y * \cos(t)), \tag{4.10b}$$

where $t = 0.4 - 6.0/(1 + x_0^2 + y_0^2)$. In this study, the parameter $\mu = 0.9$ and $x_0 = 0.1$, $y_0 = 0.1$, are the initial values for x and y respectively. The equations (4.10a) and (4.10b) are solved to create the time series $x(t)$ using the parameter and the initial conditions specified. This study focuses only on variable x of Ikeda equations.

4.3.3 Lorenz Time Series

The Lorenz (LZ) map was proposed by Lorenz [130] and is used to model the thermal driving of the convective rolls (rotating air) in the atmosphere. The LZ time series can be generated using the following three-dimensional set of first-order non-linear differential equations:

$$\dot{x} = \sigma(y - x), \tag{4.11a}$$

$$\dot{y} = rx - y - xz, \tag{4.11b}$$

$$\dot{z} = xy - bz, \tag{4.11c}$$

where the parameters σ, r and b are set, respectively, to the standard values 10, 28 and 8/3. The initial conditions for the solution of the differential equations are $x = 0.9$, $y = 0.9$ and $z = 0.9$. The equations (4.11a), (4.11b) and (4.11c) are solved to create the time series $x(t)$ using the parameters and initial conditions specified. We only focus on variable x of the Lorenz equations in our study.

4.4 Experiments and Results

After having considered the transition process from T1 to T2 FSs by presenting the FOU creation method CB, in this section, we proceed to the FLS design and evaluation process. We conduct a series of in-depth experiments using the described process in the context of three well-known time series, MG, IK and LZ, and we present their results.

Following Chapter 3 (Section 3.3) that provides a general process framework designed to systematically describe the initial design of the T1 FLS for a given application and its subsequent transformation to one or more IT2 FLSs by applying the FOU creation method, each of the process steps is detailed below and applied in the same manner to the three time series. For each, we show two cases based on different types of FSs: triangular and Gaussian MFs.

First, we generate a data set (both training and testing data) from each of the three time series. Then, we design evenly distributed T1 FSs and create the rule base by applying the WM approach using the training data set.

Next, we start the design of a series of IT2 FLSs by generating IT2 FSs using the FOU size parameter c to form the IT2 MFs of the systems. The actual number of FSs and the rules are maintained from the T1 system. In parallel, different levels of noise as a source of uncertainty are employed to generate a series of testing data sets. Finally, the performance of each of the T1/IT2 FLSs is evaluated for each of the testing data sets to determine the best mapping between the FLSs (with different FOU sizes) and the noise levels.

The complete process is illustrated by the flowchart in Figure 4.4 and can be summarised in the four subsections (4.4.1 - 4.4.4) below:

4.4.1 Data Generation

Assume a time series $x(k)$, where $k = 1,2,3,\ldots,N$. For a single stage prediction for x, we consider p past known data points of $x(k)$ to predict the future value $x(k+1)$. So, the past data of $x(k)$ time series: $x(k-p+1), x(k-p+2), x(k-p+3), \ldots, x(k)$ are used to predict the future value $x(k+1)$. Further, if these points contain uncertainty/noise, we refer to the given value of the time series $x(k)$ as $s(k)$, where $s(k) = x(k) + n(k)$ and $n(k)$ is the noise [13]. The N given time series data samples are commonly split into D training points and $(N-D)$ testing points where one usually obtains the training data points as $(x(1), x(2), x(3), \ldots, x(D))$ and testing data as $(x(D+1), x(D+2), x(D+3), \ldots, x(N))$.

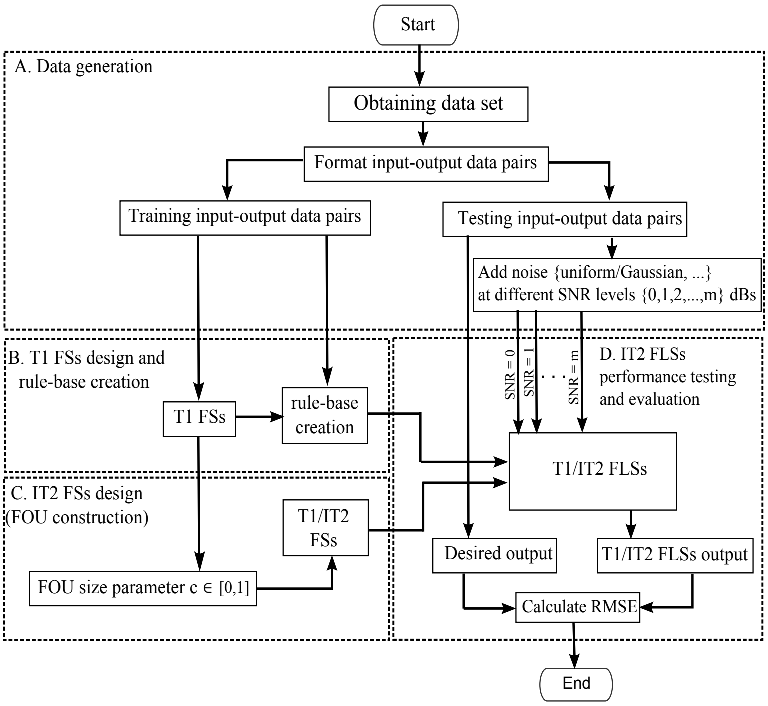


Figure 4.4 Flowchart of the process of using different FOU sizes to design and evaluate IT2 FLSs at different noise levels for the controlled blurring (CB) method.

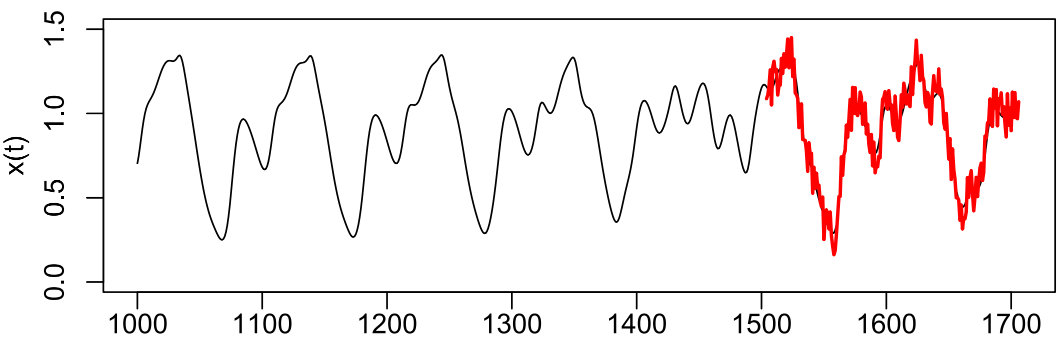
In this investigation, a single-stage prediction for the MG, IK and LZ chaotic time series is used (i.e. to predict the future value $x(k+1)$). We have considered two types of noise: uniform noise (generated from a uniform random variable) and Gaussian noise (generated from a normal distribution) (see Chapter 2, Section 2.8.1).

Following Chapter 3 (Section 3.3.1), noise-free (NF) data for the three time series are generated using (4.9) for the MG, (4.10a,4.10b) for the IK and (4.11a, 4.11b, 4.11c) for the LZ time series with the parameters and the numerical solutions of the differential equations stated above. For the IK and LZ, we use variable x in this study.

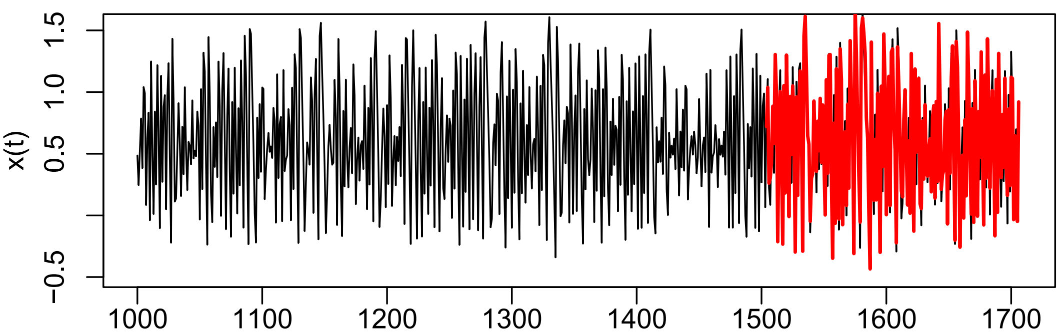
To obtain training and testing input-output data pairs, we specify the values of $N = 700$, $D = 500$ and $p = 4$. Based on these data we proceed to design a four-input, one-output T1 FLS for each of the three time series. Specifically, we extract 700 input-output data pairs as described above. The first 500 pairs (the training dataset) were used for training the FLSs (generating the rules) using $x(1001)$ to $x(1504)$, while the remaining 200 pairs (the data testing set) using $x(1505)$ to $x(1708)$ were used as the basis for testing the systems. Note that 500 pairs require 504 samples for training and 200 pairs require 204 samples for testing due to the chosen four-input FLS structure. In this work, different versions of the testing data are generated, i.e. the testing data are corrupted with two different types of noise including zero-mean uniform noise and Gaussian noise for different SNRs. We use 12 noise levels in testing for each additive noise case. Specifically, we use discretized levels from 0dB to 20dB with increments of 2, as well as the original NF data set (noise-free). In Figure 4.5, the training and testing data corrupted with additive noise (here, uniform noise) at SNR level 10dB are shown for the three time series.

4.4.2 Type-1 Fuzzy Sets Design and Rule-Base Creation

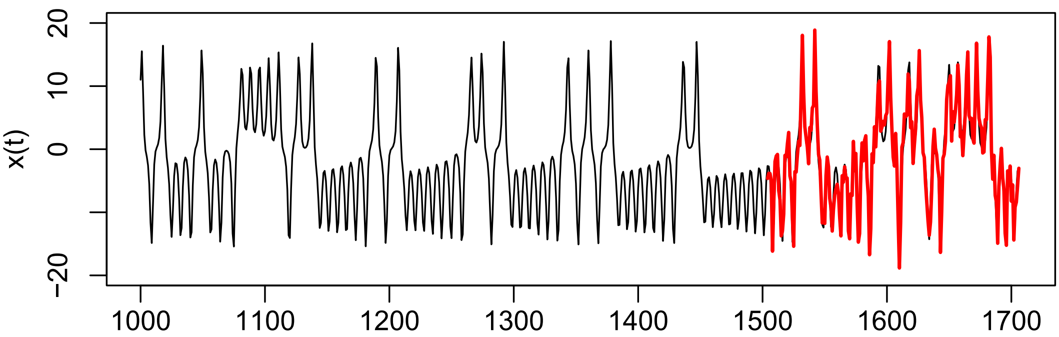
Following Chapter 3 (Section 3.3.2), the number of MFs assigned to each input and output of the FLS is chosen to be 7. While a higher number of MFs would enable better performance, 7 is a good compromise between readability (in figures) and reasonable performance, especially as optimal prediction performance is not a primary aim of this case study. First, we define the FSs to evenly cover the input and output spaces. The triangular T1 MFs used for the inputs and outputs are shown in Figure 4.6(a), whereas the Gaussian T1 MFs used for the inputs and outputs are shown in Figure 4.7(a). The MFs are labelled using numbers (e.g., F11 represents FS 1 of input 1 and for the output F1 represent FS 1). Then, we apply the WM method in order to generate the rules from the given input-output pairs (NF training data).



(a)



(b)



(c)

Figure 4.5 Training and testing data of the three time series. (a) MG time series, (b) IK time series and (c) LZ time series. Training is performed with 500 input-output pairs in $x(1001), x(1002), \ldots, x(1504)$ and testing is done with 200 input-output pairs (red lines) in $x(1505), x(1506), \ldots, x(1708)$ corrupted with additive noise (uniform noise) at SNR level 10dB.

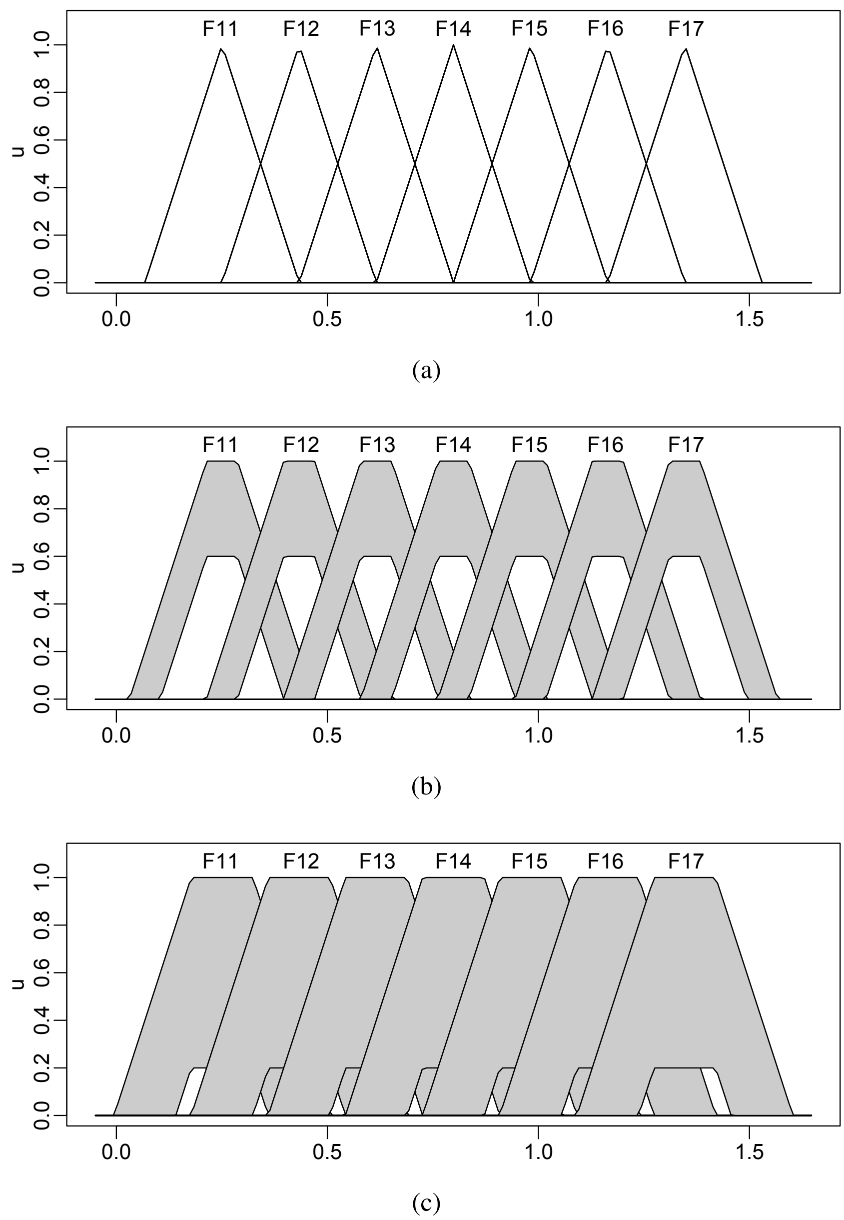


Figure 4.6 Triangular T1 and two examples of IT2 MFs used in the design of the TSP FLSs using the CB method. (a) T1 MFs, (b) IT2 MFs designed with FOU size parameter $c = 0.4$ and (c) IT2 MFs designed with FOU size parameter $c = 0.8$.

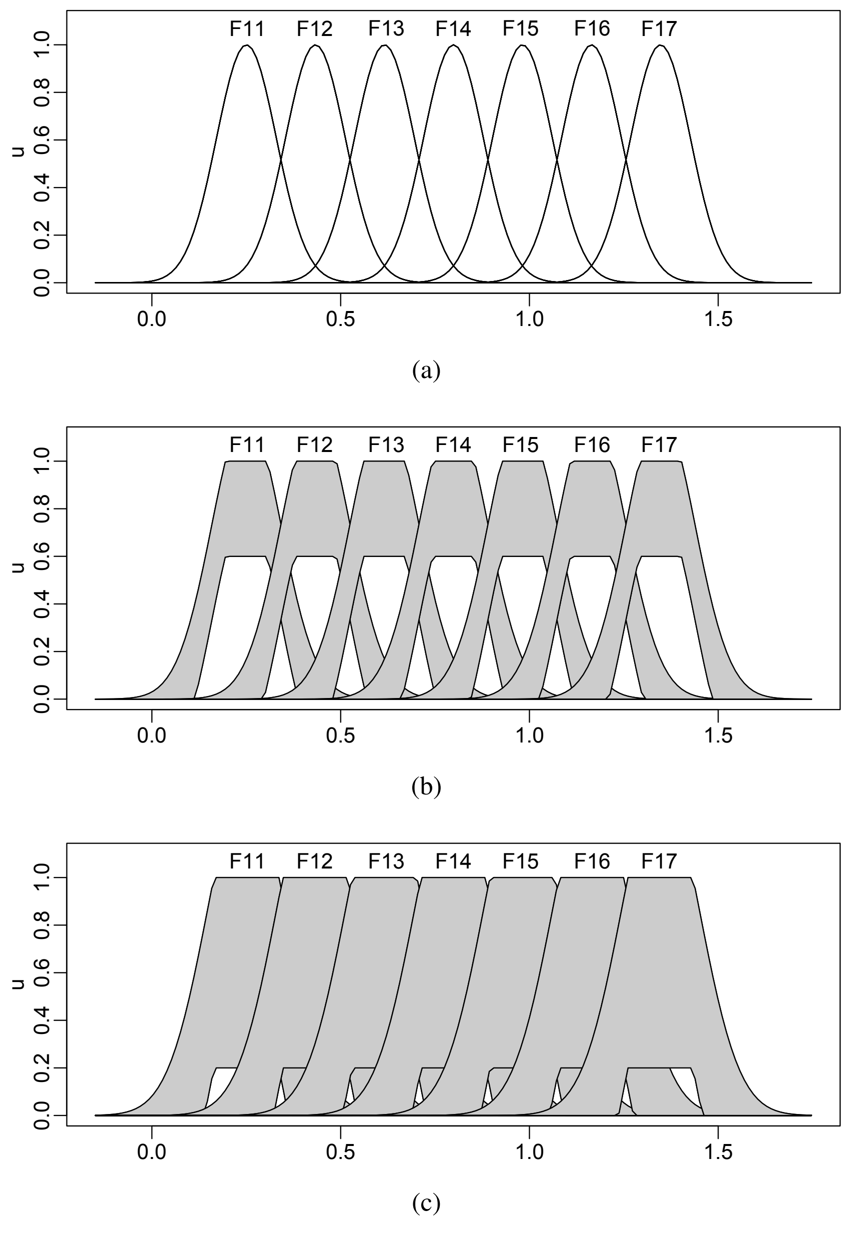


Figure 4.7 Gaussian T1 and two examples of IT2 MFs used in the design of the TSP FLSs using the CB method. (a) T1 MFs, (b) IT2 MFs designed with FOU size parameter $c = 0.3$ and (c) IT2 MFs designed with FOU size parameter $c = 0.8$.

The WM method has been popular in the field of fuzzy systems because of its simplicity and effectiveness in data-driven fuzzy rule generation. The WM method is a generic method for creating a fuzzy rule base that contains a combination of rules generated from numerical data and linguistic labels and, often, associated MFs given by expert(s). Most commonly, the input and output domain spaces are evenly distributed into a number of MFs, each associated with a respective linguistic label, before the use of the WM method. A weight is assigned to each rule generated by the method and conflicts are resolved by selecting those rules with the maximum weight. More details of the WM method can be found in Chapter 2 (Section 2.9.1).

In the current investigation, it is the FSs only that are later modified to generate different FLSs. The same rule base is employed for all FLSs in order to enable the comparison of all FLSs with a sole focus on their FSs (rather than differences in the rules). We acknowledge that this approach does not guarantee the best rule base for each individual FLS; however, as the aim of the study is a comparison based on the underling FSs rather than achieving best performance per se, we believe this approach is suitable.

The number of rules generated using WM methods for each time series using NF training data set are 47 rules for MG, 192 rules for IK and 70 rules for LZ. The resulting rules are used for all the FLSs in our experiments in order to enable a comparison that focuses on the FSs.

4.4.3 Interval Type-2 Fuzzy Sets Creation

Following Chapter 3 (Section 3.3.3), we extend the T1 FLS into a series of IT2 FLSs using the partially dependent design approach. First, the FOU size parameter $c \in [0, 1]$

is discretized to a set of 11 values starting at 0 and increasing to a maximum of 1.0 in increments of 0.1. If the FOU size parameter $c = 0.0$, the IT2 FSs reduce to the original T1 FSs, whereas in the case of using $c = 1.0$, the IT2 FSs reach the maximum amount of their width (i.e. the FOU covers the entire primary membership). Thus, we design 11 IT2 FLSs, where each system uses specific IT2 FSs with the given FOU size parameter c.

To construct the UMF and LMF of the IT2 FSs, we use the controlled blurring (CB) method detailed in Section 4.2 and apply equations (4.1) and (4.2) by combining the T1 FSs with the chosen FOU size represented by the FOU size parameter c. Examples of the FOU construction for the FSs of two IT2 FLSs using different FOU size parameters $c = 0.40$ and $c = 0.80$ are shown in Figure 4.6(b) and Figure 4.6(c) for the triangular case, and examples of the Gaussian case are shown in Figure 4.7(b) and Figure 4.7(c).

4.4.4 Fuzzy Logic Systems Evaluation

After finishing the design of the IT2 FLSs with the chosen FOU sizes, the testing data sets are used to test and evaluate the performance of the individual IT2 FLSs when faced with different uncertainty/noise levels. Now that we have 11 FLSs, each using different FOU sizes determined by the given c, we test each of the FLSs against 12 levels of noise in order to determine which FOU size results in the best performance for each given noise level. Each test is repeated 30 times to account for the random generation of the noise.

The performances of all the designs were evaluated using their RMSE based on (3.2), i.e.,

$$RMSE = \sqrt{\frac{1}{200}\sum_{k=1508}^{1707}\left[s(k+1) - f\left(\mathbf{s}^{(k)}\right)\right]^2} \tag{4.12}$$

where, $s(k+1)$ is the output of the noisy testing data, $f(s^{(k)})$ is the crisp output of the FLS and $\mathbf{s}^{(k)} = [s(k-3), s(k-2), s(k-1), s(k)]^T$.

The RMSE results are averaged over 30 runs and are depicted in Tables 4.1-4.6 showing the results of the three time series using the two different MF types (triangular and Gaussian). Each column represents an IT2 FLS design with a given FOU size parameter c. The rows show the average RMSE values for both uniform noise and Gaussian noise (shaded values) at the different SNR values for all FOU sizes/FLSs. The bold values are the minimum of each average RMSE representing the best FLS (based on the respective FOU size) at a particular SNR level for both uniform (non-shaded) and Gaussian (shaded) noise respectively.

4.5 Results Analysis and Discussion

In this section, the experiment results are analysed and discussed. First, we focus on the relationship between the FOU size, uncertainty/noise levels and FLS performance. Then, we discuss the possibility of determining optimal FOU sizes based on known levels of uncertainty (SNR).

Table 4.1 Average RMSE values for MG TSP for triangular MFs for both uniform (non-shaded) and Gaussian (shaded) noise. The bold values highlight the row's (i.e., c value and thus FOU size) minimum RMSE at each noise/SNR level.

	c=0	c=0.1	c=0.2	c=0.3	c=0.4	c=0.5	c=0.6	c=0.7	c=0.8	c=0.9	c=1.0
NF	**0.03128**	0.03135	0.03183	0.03347	0.03689	0.04145	0.04694	0.05309	0.05926	0.06576	0.07844
	0.03128	0.03135	0.03183	0.03347	0.03689	0.04145	0.04694	0.05309	0.05926	0.06576	0.07844
20	0.04257	0.04242	**0.04233**	0.04305	0.04515	0.04834	0.05198	0.05591	0.06103	0.06884	0.07651
	0.0418	**0.04176**	0.04195	0.04292	0.04491	0.04771	0.05119	0.05561	0.06225	0.07241	0.07679
18	0.04803	0.04785	**0.04765**	0.04813	0.04973	0.05218	0.05516	0.05887	0.06375	0.07063	0.07860
	0.04747	**0.04737**	0.04739	0.04801	0.04940	0.05145	0.05430	0.05819	0.065	0.07449	0.07779
16	0.05537	0.05519	**0.05492**	0.05517	0.05630	0.05798	0.06011	0.06321	0.06712	0.07369	0.08002
	0.05507	0.05496	**0.05489**	0.05522	0.05601	0.05721	0.05902	0.06248	0.06864	0.07745	0.07999
14	0.06504	0.06484	**0.06453**	0.06461	0.06526	0.06620	0.06735	0.06916	0.07221	0.07815	0.08507
	0.06642	0.06495	**0.06474**	0.06486	0.06523	0.06556	0.06618	0.06854	0.07412	0.08253	0.08409
12	0.07762	0.07738	0.07693	**0.07669**	0.07690	0.07745	0.07801	0.07866	0.08041	0.08505	0.09238
	0.084	0.0796	0.0793	0.07755	0.07738	0.07675	**0.07631**	0.07735	0.08214	0.09001	0.09125
10	0.09970	0.09560	0.09272	0.09215	0.09175	0.09141	0.09115	**0.09105**	0.09232	0.09668	0.10240
	0.10101	0.09806	0.09725	0.09552	0.09416	0.09211	**0.08963**	0.08973	0.09332	0.10139	0.10353
8	0.15123	0.12600	0.11441	0.11154	0.10884	0.10720	0.10638	**0.10605**	0.10778	0.11201	0.11597
	0.1353	0.12831	0.12077	0.11502	0.1125	0.11016	0.10811	**0.10664**	0.11019	0.11791	0.11872
6	0.23282	0.20347	0.17273	0.15102	0.13709	0.12751	0.12381	**0.12268**	0.12343	0.12677	0.13167
	0.17548	0.16345	0.15316	0.14389	0.13903	0.13269	0.12852	**0.12762**	0.13085	0.13863	0.13943
4	0.33545	0.29805	0.26385	0.22440	0.19338	0.17536	0.16009	0.15032	**0.14304**	0.14512	0.14874
	0.23802	0.2157	0.19374	0.17844	0.16717	0.16023	0.15507	**0.15431**	0.15637	0.163	0.16188
2	0.42967	0.38805	0.35537	0.32050	0.28663	0.25452	0.22955	0.20562	0.19640	0.18152	**0.17779**
	0.29494	0.2701	0.2526	0.23175	0.21649	0.20201	0.19435	**0.1846**	0.18696	0.1912	0.19249
0	0.51499	0.47755	0.44177	0.40416	0.37143	0.34207	0.31091	0.28695	0.26323	0.24890	**0.22753**
	0.36322	0.33805	0.31348	0.2886	0.27234	0.25012	0.23508	0.2245	0.22092	0.22368	**0.22075**

4.5.1 Relationship Between FOU Size, Uncertainty Levels and FLS Performance

The presented experiments illustrate the relationships between FOU size and the uncertainty/noise levels in the three case studies of the MG, IK and LZ time series. From Tables 4.1-4.6, a direct relationship between the FOU size, the SNR and the performance can be summarised in the sense that as the uncertainty/noise level increases (SNR value decreases), the FLSs with increasing FOU size achieve better performance

Table 4.2 Average RMSE values for MG TSP for Gaussian MFs for both uniform (non-shaded) and Gaussian (shaded) noise. The bold values highlight the row's (i.e., c value and thus FOU size) minimum RMSE at each noise/SNR level.

	c=0	c=0.1	c=0.2	c=0.3	c=0.4	c=0.5	c=0.6	c=0.7	c=0.8	c=0.9	c=1.0
NF	**0.03325**	0.03344	0.03439	0.03649	0.03966	0.04371	0.04812	0.05315	0.05997	0.06922	0.07536
	0.03325	0.03344	0.03439	0.03649	0.03966	0.04371	0.04812	0.05315	0.05997	0.06922	0.07536
20	0.04243	**0.04243**	0.04281	0.04389	0.04582	0.04850	0.05187	0.05603	0.06156	0.06899	0.07416
	0.04217	0.04188	**0.04174**	0.04191	0.04296	0.04511	0.04806	0.05182	0.0568	0.06283	0.06749
18	0.04720	**0.04719**	0.04745	0.04818	0.04949	0.05154	0.05428	0.05784	0.06268	0.06934	0.07384
	0.04745	0.0472	**0.04697**	0.04698	0.04763	0.04909	0.0513	0.0544	0.05873	0.06388	0.06811
16	0.05378	**0.05378**	0.05394	0.05429	0.05494	0.05606	0.05793	0.06071	0.06462	0.07032	0.07471
	0.05465	0.05448	0.05417	**0.05400**	0.05418	0.0548	0.05606	0.05819	0.06143	0.06543	0.06873
14	0.06256	0.06260	0.06263	0.06262	**0.06254**	0.062570	0.06326	0.06495	0.06776	0.07236	0.07620
	0.06414	0.06409	0.06375	**0.0633**	0.06304	0.0628	0.06287	0.06347	0.06506	0.06776	0.07062
12	0.07409	0.07414	0.07391	0.07341	0.0726	0.07149	**0.07074**	0.07098	0.07244	0.07578	0.07858
	0.07689	0.07694	0.07635	0.07556	0.07457	0.0732	0.07186	0.07086	**0.07084**	0.07213	0.07421
10	0.08900	0.08894	0.08823	0.08699	0.08523	0.08305	0.08085	**0.07941**	0.07946	0.08115	0.08299
	0.09376	0.09378	0.09265	0.09105	0.08904	0.08652	0.08375	0.08131	0.07977	**0.07956**	0.08059
8	0.10750	0.10717	0.10574	0.10349	0.10068	0.09748	0.09409	0.09107	0.08929	**0.08909**	0.08958
	0.11422	0.1141	0.11225	0.10947	0.10604	0.1023	0.09872	0.09543	0.09264	0.09056	**0.09021**
6	0.12853	0.12800	0.12579	0.12246	0.1185	0.1144	0.11018	0.10597	0.10258	0.10041	**0.09979**
	0.13737	0.13713	0.13416	0.1302	0.12552	0.12082	0.11663	0.11271	0.10899	0.10526	**0.10354**
4	0.15170	0.15095	0.14797	0.14381	0.13903	0.13397	0.12893	0.12399	0.11961	0.11616	**0.11471**
	0.16516	0.16419	0.16022	0.15501	0.14917	0.14339	0.13808	0.13342	0.12907	0.12395	**0.12128**
2	0.17931	0.17792	0.17373	0.16861	0.16266	0.15688	0.15136	0.14601	0.14125	0.13712	**0.13505**
	0.20022	0.19768	0.1919	0.18545	0.17832	0.17111	0.16429	0.15816	0.15246	0.14583	**0.1402**
0	0.21296	0.20979	0.20431	0.19808	0.1915	0.18513	0.17887	0.17311	0.16777	0.16165	**0.15515**
	0.24429	0.24012	0.23197	0.22327	0.21437	0.20542	0.19647	0.1879	0.17923	0.16755	**0.15908**

(based on minimum RMSE value). While this result is intuitive, it is valuable to establish that it is *solely* the controlled blurring of a type-1 FLS (trained in a NF environment) that has enabled the designed IT2 FLSs to achieve better performance under the different levels of noise – the key is to establish the appropriate size of the FOU for the given noise levels.

The first column of these tables contains the performance results of the IT2 FLS designed using FOU size parameter $c = 0.0$ which reduces to the original T1 FLS.

Table 4.3 Average RMSE values for IK TSP for triangular MFs for both uniform (non-shaded) and Gaussian (shaded) noise. The bold values highlight the row's (i.e., c value and thus FOU size) minimum RMSE at each noise/SNR level.

	c=0	c=0.1	c=0.2	c=0.3	c=0.4	c=0.5	c=0.6	c=0.7	c=0.8	c=0.9	c=1.0
NF	**0.26957**	0.27303	0.27619	0.28602	0.30833	0.33907	0.36813	0.39895	0.42892	0.45229	0.46784
	0.26957	0.27303	0.27619	0.28602	0.30833	0.33907	0.36813	0.39895	0.42892	0.45229	0.46784
20	0.29715	**0.29622**	0.29726	0.30525	0.32111	0.3436	0.37148	0.40122	0.43079	0.45522	0.46894
	0.30218	**0.29727**	0.29838	0.30426	0.31748	0.33873	0.36505	0.39377	0.42098	0.44438	0.45794
18	0.31186	0.31081	**0.31074**	0.31596	0.3281	0.34841	0.37386	0.40265	0.43117	0.45509	0.46884
	0.31937	0.31338	**0.31331**	0.31718	0.32677	0.34416	0.36782	0.39504	0.4217	0.44392	0.45731
16	0.3325	0.33035	**0.32864**	0.33045	0.34017	0.35594	0.37827	0.40497	0.43204	0.4546	0.46881
	0.34294	0.33673	**0.33472**	0.33492	0.34045	0.35388	0.37381	0.39805	0.42239	0.44374	0.45726
14	0.36277	0.35877	0.35192	**0.35062**	0.35521	0.36754	0.38644	0.40977	0.43397	0.45437	0.4672
	0.38088	0.37487	0.36668	**0.36323**	0.36497	0.37272	0.38593	0.40512	0.42506	0.44352	0.45625
12	0.40235	0.39494	0.38523	0.37887	**0.37789**	0.38384	0.39722	0.41695	0.43703	0.45464	0.46681
	0.42707	0.4201	0.4097	0.39949	**0.39563**	0.3986	0.40565	0.41827	0.4351	0.45052	0.46035
10	0.44828	0.4389	0.42485	0.41297	**0.40616**	0.40849	0.41907	0.43076	0.44467	0.45867	0.46934
	0.47581	0.46767	0.45294	0.4375	0.42865	**0.42717**	0.43101	0.43993	0.45049	0.46341	0.4717
8	0.49325	0.48471	0.46806	0.45464	0.44434	**0.44199**	0.44307	0.45103	0.46243	0.47218	0.48123
	0.53066	0.52096	0.50514	0.48759	0.47099	0.46344	**0.46147**	0.46643	0.47526	0.48313	0.48629
6	0.54808	0.53923	0.52008	0.50292	0.48959	0.47738	**0.47534**	0.4769	0.48324	0.49234	0.49827
	0.61009	0.59946	0.58313	0.56096	0.54192	0.52752	0.5183	**0.51308**	0.51422	0.51606	0.51672
4	0.61675	0.60696	0.58597	0.56424	0.54485	0.52447	0.51233	0.50661	**0.50600**	0.50959	0.51553
	0.67866	0.67583	0.65263	0.63426	0.61275	0.59574	0.57675	0.56626	0.55827	0.55443	**0.55437**
2	0.6928	0.67793	0.65804	0.63253	0.60816	0.58381	0.56660	0.55111	0.54507	0.54555	**0.54185**
	0.76266	0.76508	0.74538	0.71579	0.68221	0.66153	0.64098	0.62175	0.6115	0.60444	**0.59774**
0	0.76955	0.75801	0.73781	0.71321	0.68411	0.65713	0.63336	0.61341	0.5989	0.58885	**0.57991**
	0.86766	0.86724	0.84402	0.81152	0.77899	0.74138	0.70517	0.68294	0.6694	0.65563	**0.64393**

From Tables 4.1-4.6, it is clear that T1 FLSs are outperforming IT2 in NF testing data, as expected.

In studying the results in the above tables, we see that as the noise level increases (moving down in the tables), the best results occur as the FOU size increases (moving to the right in the tables). However, as expected, performance degradation is recorded at higher levels of noise. For example, in the MG triangular case, significant performance degradation starts at 6dBs, as shown in Figure 4.8(a) and Table 4.1, where the RMSE starts to exceed 0.20 in the first 2 FLSs (i.e. at FOU sizes, 0.0 and 0.10) and then

Table 4.4 Average RMSE values for IK TSP for Gaussian MFs for both uniform (non-shaded) and Gaussian (shaded) noise. The bold values highlight the row's (i.e., c value and thus FOU size) minimum RMSE at each noise/SNR level.

	c=0	c=0.1	c=0.2	c=0.3	c=0.4	c=0.5	c=0.6	c=0.7	c=0.8	c=0.9	c=1.0
NF	**0.26294**	0.26567	0.26911	0.27950	0.30163	0.32803	0.35765	0.38543	0.41124	0.43379	0.45238
	0.26294	0.26567	0.26911	0.27950	0.30163	0.32803	0.35765	0.38543	0.41124	0.43379	0.45238
20	0.29013	**0.28859**	0.29148	0.30062	0.31706	0.33967	0.36458	0.38968	0.41320	0.43541	0.45376
	0.28762	**0.28612**	0.28855	0.29637	0.31088	0.33124	0.35484	0.37918	0.40249	0.42496	0.44222
18	0.30764	**0.30515**	0.30533	0.31183	0.32572	0.34541	0.36850	0.39256	0.41516	0.43649	0.45353
	0.30642	0.30378	**0.30352**	0.30887	0.32109	0.339	0.36015	0.38281	0.40468	0.4258	0.44245
16	0.33055	0.32722	**0.32520**	0.32803	0.33770	0.35426	0.37465	0.39672	0.41807	0.43802	0.45351
	0.33153	0.32848	**0.32550**	0.3273	0.33593	0.35053	0.36867	0.38853	0.40818	0.4273	0.44267
14	0.35848	0.35481	0.34990	**0.34895**	0.35451	0.36663	0.38389	0.40321	0.42213	0.44003	0.45423
	0.3687	0.36576	0.35921	**0.35657**	0.35977	0.36899	0.38212	0.39755	0.41346	0.42974	0.44333
12	0.39095	0.38697	0.37928	**0.37450**	0.37519	0.38339	0.39609	0.41166	0.42751	0.44289	0.45548
	0.41372	0.40901	0.39956	0.39158	**0.38928**	0.39237	0.40022	0.41087	0.4227	0.43558	0.44668
10	0.42940	0.42602	0.41354	0.40445	**0.40113**	0.40390	0.41200	0.42281	0.43480	0.44716	0.45783
	0.45911	0.45262	0.43845	0.42717	0.41987	**0.41734**	0.41983	0.4262	0.43445	0.44404	0.45287
8	0.47579	0.47124	0.45566	0.44181	0.43198	**0.42848**	0.43027	0.43694	0.44463	0.45357	0.46205
	0.51198	0.50216	0.48489	0.4677	0.4542	0.44724	**0.44446**	0.44553	0.44977	0.45555	0.46202
6	0.53152	0.52397	0.50399	0.48395	0.46872	0.45824	0.45344	**0.45332**	0.45695	0.46314	0.46883
	0.59391	0.58367	0.55917	0.53523	0.51478	0.49694	0.48479	0.47828	**0.47582**	0.47689	0.47946
4	0.59681	0.58784	0.56169	0.53538	0.51089	0.49271	0.48095	0.47507	**0.47449**	0.47690	0.48014
	0.67303	0.66267	0.63692	0.60558	0.57564	0.55087	0.53023	0.51579	0.50658	0.50248	**0.50215**
2	0.67404	0.65842	0.62764	0.59663	0.56615	0.53874	0.51820	0.50526	0.49871	**0.49632**	0.49632
	0.76597	0.753	0.72459	0.68684	0.64935	0.61357	0.58164	0.55644	0.53979	0.53054	**0.52697**
0	0.75918	0.74135	0.70713	0.66952	0.63139	0.59699	0.56738	0.54397	0.52794	0.51727	**0.50994**
	0.88076	0.86183	0.82146	0.77594	0.72808	0.68126	0.63972	0.6034	0.57818	0.56182	**0.55479**

decreases again until reaching its minimum at $c = 0.70$. Overall, as the level of uncertainty/noise increases (i.e. decreasing SNR), it is increasingly only FLSs with wider FOUs (higher c) that achieve reasonable performance.

Figures 4.9, 4.10 and 4.11 illustrate the results of the MG, IK and LZ TSP respectively, using IT2 FLS with different FOU sizes ($c = 0.2$ and $c = 0.6$) at different noise levels (20dB and 10dB). From these figures and as a general trend, as the FOU size increases, the performance increases as well.

Table 4.5 Average RMSE values for LZ TSP for triangular MFs for both uniform (non-shaded) and Gaussian (shaded) noise. The bold values highlight the row's (i.e., c value and thus FOU size) minimum RMSE at each noise/SNR level.

	c=0	c=0.1	c=0.2	c=0.3	c=0.4	c=0.5	c=0.6	c=0.7	c=0.8	c=0.9	c=1.0
NF	**1.9038**	1.9072	1.9209	1.9565	2.0246	2.1444	2.3169	2.558	2.9614	3.3302	3.7637
	1.9038	1.9072	1.9209	1.9565	2.0246	2.1444	2.3169	2.558	2.9614	3.3302	3.7637
20	2.2904	2.2874	**2.284**	2.2921	2.3286	2.4256	2.6227	2.9516	3.3546	3.753	4.068
	2.2445	2.2415	**2.239**	2.2497	2.2907	2.3922	2.6014	2.9429	3.3524	3.7454	4.0391
18	2.4748	2.4695	2.4594	**2.4571**	2.4877	2.5804	2.7825	3.1128	3.4935	3.8535	4.1097
	2.4482	2.4429	2.4334	**2.4331**	2.4622	2.559	2.7765	3.1158	3.5056	3.8558	4.1374
16	2.7343	2.7264	2.7111	**2.7026**	2.7201	2.817	3.007	3.319	3.6528	3.9617	4.2165
	2.7426	2.7345	2.7168	**2.7053**	2.7213	2.8175	3.0247	3.3474	3.6937	4.0201	4.2412
14	3.0944	3.0847	3.0615	**3.0429**	3.062	3.1426	3.3088	3.5573	3.8537	4.1543	4.3299
	3.3398	3.3281	3.302	**3.2847**	3.297	3.3609	3.5381	3.8011	4.0799	4.335	4.4781
12	3.5932	3.5789	3.5472	3.5217	**3.5155**	3.5577	3.6778	3.868	4.1066	4.3465	4.5313
	4.2023	4.187	4.1341	4.0905	**4.0701**	4.0939	4.2068	4.3703	4.555	4.6958	4.7889
10	4.2619	4.2419	4.1982	4.1386	4.1033	**4.0998**	4.1525	4.2827	4.4513	4.6175	4.768
	5.2908	5.2708	5.2085	5.1171	5.0466	5.0133	**5.0041**	5.062	5.1458	5.1754	5.1873
8	5.0869	5.0606	5.0025	4.9289	4.8514	**4.8002**	4.8148	4.8645	4.9189	5.0147	5.109
	6.4871	6.4599	6.3796	6.295	6.1831	6.0616	5.9673	5.9218	5.8533	5.7716	**5.7276**
6	6.1011	6.0691	5.9991	5.8916	5.7619	5.6701	5.6226	5.618	5.624	5.6004	**5.5957**
	7.7795	7.7631	7.6746	7.5565	7.4017	7.1955	7.0527	6.882	6.7169	6.5246	**6.382**
4	7.3221	7.2876	7.1722	7.0485	6.8828	6.7265	6.6675	6.5688	6.4763	6.4219	**6.4071**
	9.2143	9.1973	9.0545	8.8525	8.6504	8.4238	8.2258	7.9935	7.7641	7.4842	**7.2411**
2	8.6899	8.6892	8.565	8.386	8.2147	8.0435	7.8912	7.7669	7.6665	7.5192	**7.412**
	10.762	10.753	10.597	10.347	10.084	9.7842	9.5332	9.3026	9.0351	8.7395	**8.4371**
0	10.275	10.245	10.125	9.9854	9.8153	9.6029	9.4177	9.2714	9.0839	8.7285	**8.4439**
	12.724	12.761	12.567	12.371	12.088	11.731	11.432	11.139	10.774	10.374	**10.07**

For a better illustration of the results in Tables (4.1-4.6), we show a visual representation of the these results in Figure 4.8. This figure shows the RMSE values of 11 FLSs using different noise levels (different SNR values) of testing data corrupted with uniform noise for MG, IK and LZ time series. This figure, together with Tables (4.1-4.6), shows the performance of each FLS designed with a particular FOU size (parameter c), as each is tested with different noise levels using both uniform and Gaussian noise. As can be seen from the tables, the results for Gaussian noise are

Table 4.6 Average RMSE values for LZ TSP for Gaussian MFs for both uniform (non-shaded) and Gaussian (shaded) noise. The bold values highlight the row's (i.e., *c* value and thus FOU size) minimum RMSE at each noise/SNR level.

	c=0	*c*=0.1	*c*=0.2	*c*=0.3	*c*=0.4	*c*=0.5	*c*=0.6	*c*=0.7	*c*=0.8	*c*=0.9	*c*=1.0
NF	**1.959**	1.9601	1.9683	1.9988	2.0682	2.1866	2.379	2.6781	3.1463	3.7683	4.3556
	1.959	1.9601	1.9683	1.9988	2.0682	2.1866	2.379	2.6781	3.1463	3.7683	4.3556
20	2.4153	2.4032	**2.3955**	2.3986	2.4359	2.5394	2.7368	3.0406	3.4873	4.0173	4.5568
	2.3585	**2.35**	2.3526	2.3727	2.4359	2.5811	2.8407	3.2153	3.6996	4.2108	4.5989
18	2.6147	2.6005	2.5882	**2.5831**	2.6151	2.7101	2.8955	3.1895	3.6084	4.1011	4.61
	2.5693	2.5581	**2.5534**	2.5632	2.6154	2.7515	3.0015	3.3611	3.8135	4.299	4.6656
16	2.8867	2.8706	2.8534	**2.844**	2.8641	2.9499	3.125	3.3983	3.7707	4.2132	4.6696
	2.8721	2.8574	2.8444	**2.842**	2.8813	3.0041	3.2348	3.5684	3.9738	4.4161	4.7505
14	3.2585	3.2404	3.2224	**3.2048**	3.2153	3.2879	3.4325	3.6706	3.9823	4.3564	4.7424
	3.4968	3.4778	3.4569	**3.4435**	3.4657	3.5586	3.7416	3.9999	4.2945	4.6009	4.8404
12	3.7663	3.7494	3.7266	3.7047	**3.7000**	3.7313	3.8296	4.0101	4.2445	4.5285	4.8274
	4.3521	4.331	4.3032	4.2721	**4.2543**	4.2842	4.3796	4.5231	4.6586	4.7649	4.9139
10	4.428	4.4137	4.3831	4.3432	4.3043	**4.2894**	4.3282	4.432	4.581	4.7439	5.0632
	5.397	5.376	5.3276	5.2557	5.1814	5.1258	5.1065	5.0949	5.0298	**4.9244**	4.987
8	5.245	5.231	5.186	5.1204	5.0448	4.9709	**4.9429**	4.956	4.9848	5.0183	5.3981
	6.5417	6.5177	6.4457	6.3365	6.1931	6.0412	5.8862	5.6893	5.397	5.1126	**5.0995**
6	6.2263	6.2154	6.1415	6.0327	5.9125	5.7685	5.6438	5.5232	5.4417	**5.3863**	5.8156
	7.7605	7.7333	7.6203	7.4511	7.2306	6.9695	6.6679	6.2847	5.8123	5.3893	**5.3012**
4	7.3907	7.3653	7.2443	7.0582	6.8526	6.6121	6.4003	6.1969	5.9741	5.8374	**5.7205**
	9.0454	9.0149	8.8476	8.598	8.2766	7.9118	7.4984	6.9704	6.3573	5.8184	**5.6461**
2	8.7792	8.6967	8.4426	8.169	7.8562	7.544	7.2395	6.9248	6.5357	6.3007	**6.1603**
	10.46	10.43	10.186	9.8547	9.4728	9.0307	8.4905	7.8354	7.1128	6.4695	**6.2028**
0	10.41	10.184	9.792	9.3451	8.9092	8.467	8.0241	7.5418	7.0247	6.7793	**6.6741**
	12.401	12.336	12.024	11.625	11.156	10.609	9.9559	9.1445	8.2619	7.4772	**7.0205**

consistent with the results of uniform noise and both lead to the same observations and conclusions detailed further below.

The noise levels start with NF level (no noise added to the testing data) and are increased, starting with SNR = 20db (the lowest SNR beyond NF used in this study) to SNR = 0db (the highest noise level). It is a common trend in all of these results that, as we increase the noise level, the FLSs with wider FOU sizes show improvement in their performance. This indicates a strong relationship in terms of performance between the FOU size and the noise level. These results are supported by other works,

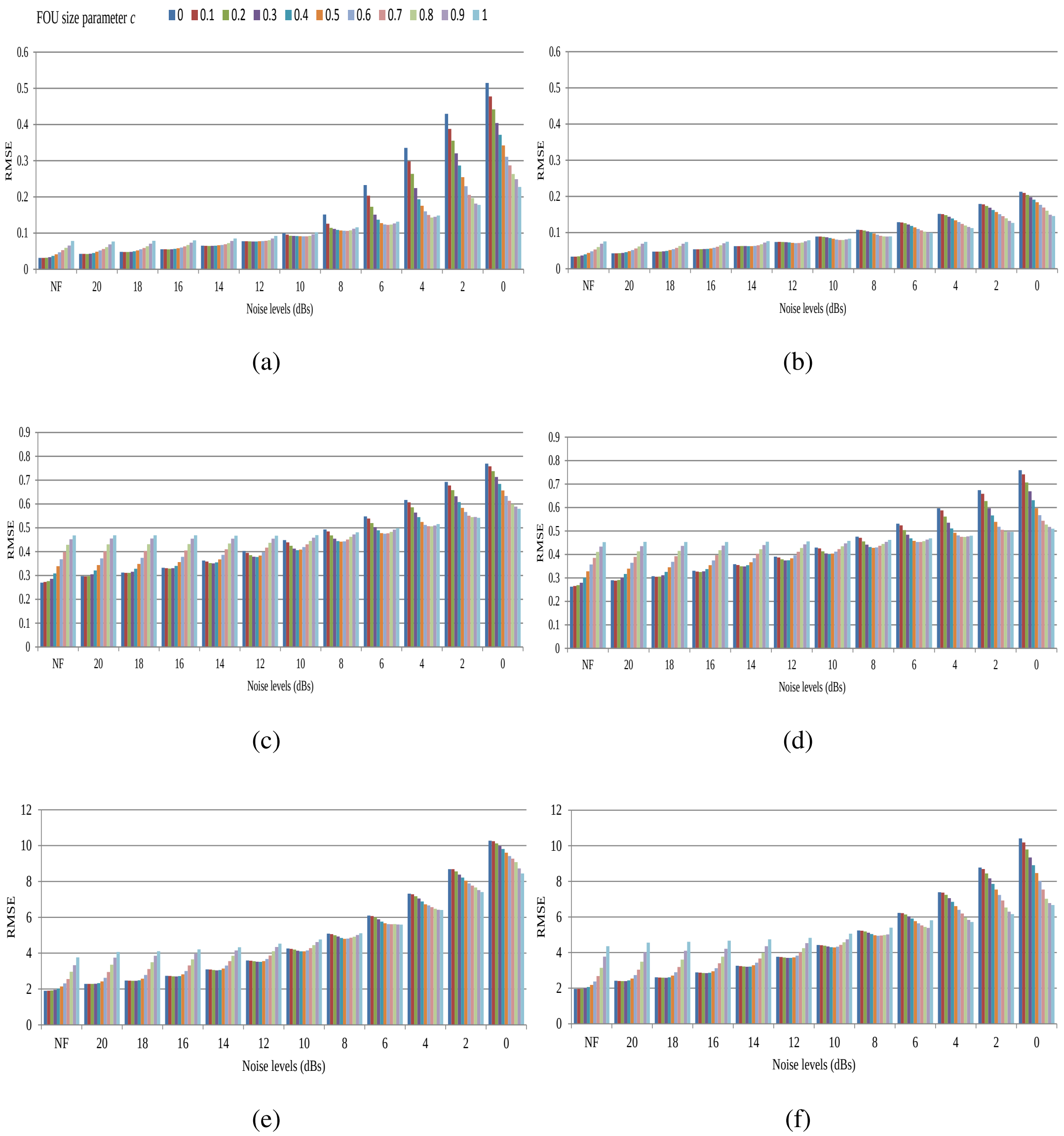


Figure 4.8 The RMSE values of 11 FLSs using testing data corrupted with uniform noise at 12 SNR levels. (a,b), (c,d) and (e,f), show the MG, IK and LZ time series results using triangular and Gaussian MFs respectively.

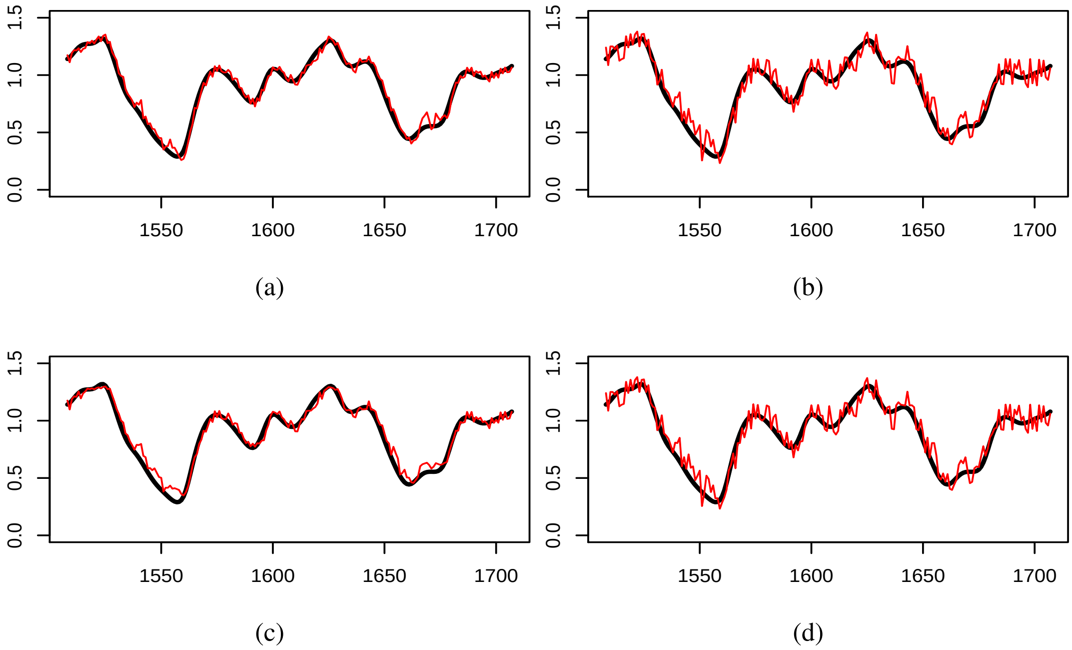


Figure 4.9 MG TSP using IT2 FLS with two FOU sizes at two noise levels. (a) $c = 0.2$ at 20dB, (b) $c = 0.2$ at 10dB, (c) $c = 0.6$ at 20dB and (d) $c = 0.6$ at 10dB.

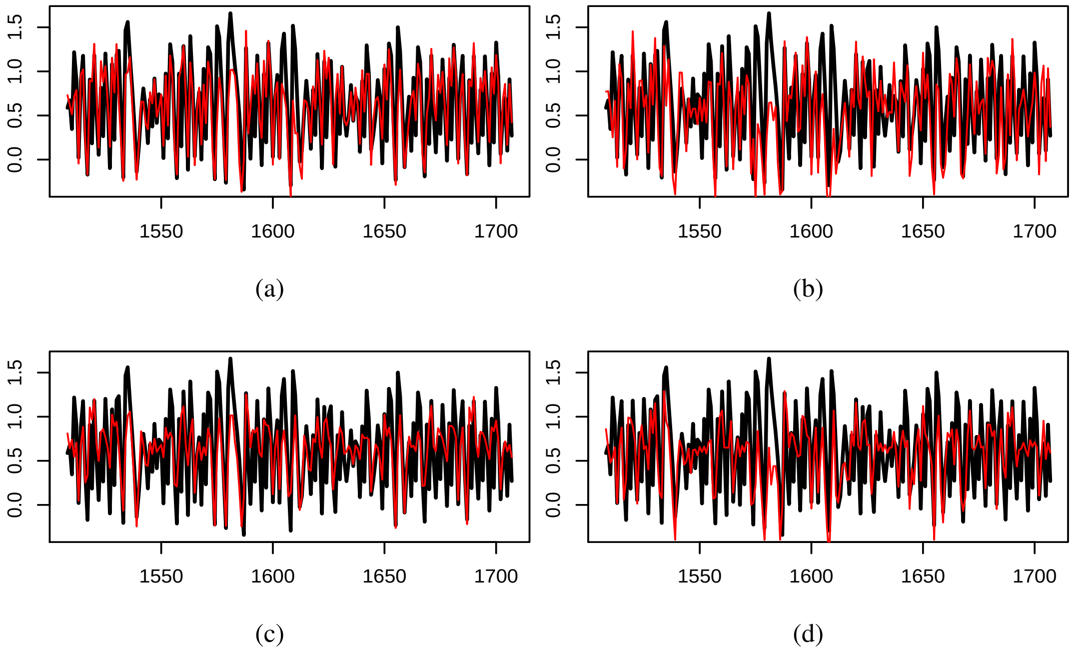
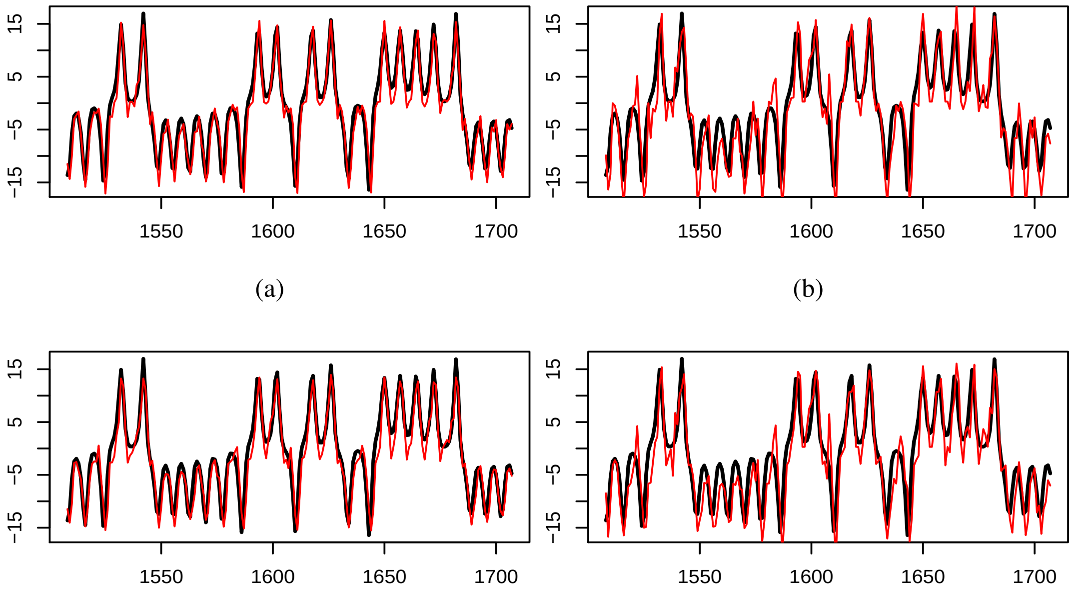


Figure 4.10 IK TSP using IT2 FLS with two FOU sizes at two noise levels. (a) $c = 0.2$ at 20dB, (b) $c = 0.2$ at 10dB, (c) $c = 0.6$ at 20dB and (d) $c = 0.6$ at 10dB.



(a) (b)

(c) (d)

Figure 4.11 LZ TSP using IT2 FLS with two FOU sizes at two noise levels. (a) $c = 0.2$ at 20dB, (b) $c = 0.2$ at 10dB, (c) $c = 0.6$ at 20dB and (d) $c = 0.6$ at 10dB.

such as the work by Khanesar et al [110], which showed that as more noise is added to the system, MFs with increased width show an improvement in the performance of IT2 FLSs over T1 FLSs.

In other words, the best performance is achieved by transitioning from the T1 FLS (FOU size parameter $c = 0.0$) to the IT2 FLS (FOU size parameter $c = 1.0$), in a continuous fashion based on the noise level(s) that the FLS is facing.

This insight follows intuition and indicates that exploiting the relationship between encountered levels of uncertainty/noise and FOU size may lead to a better choice of the FOU size and hence FLS in applications where the noise level can be quantified using a SNR or similar means. Further, the results show that in applications, it is possible to achieve improved performance *solely* by the controlled blurring of T1 MFs based on an appropriate ‘blurring level’ (in our case: c). Further, the results provide

Table 4.7 Example optimal FOU sizes collected from the RMSE tables (4.1-4.6) for uniform noise.

SNR	The optimal FOU size c for different IT2 FLSs		
	Triangular MG TSP	Triangular IK TSP	Triangular LZ TSP
NF	0	0	0
20	0.2	0.1	0.2
16	0.2	0.2	0.3
12	0.3	0.4	0.4
10	0.7	0.4	0.5
6	0.7	0.6	1.0
0	1.0	1.0	1.0

general insight into the fundamental choice between T1 and T2 FSs in applications where an assessment of the noise level (SNR) can be made or is available.

4.5.2 Determining Optimal FOU Sizes Based on Known Levels of Uncertainty

Based on the experiments conducted, we can summarize the optimal FOU sizes for the given experimental settings. Table 4.7 shows the optimal FOU size c at a set of uncertainty levels (SNRs) for chosen TSP cases based on the results shown in the Tables (4.1-4.6) using uniform noise results. For example, for a given SNR of 10db, the FOU size c for the time series MG, IK and LZ should be 0.7, 0.4 and 0.5 respectively. However, this example is for a specific application with a specific MF and one noise type (uniform noise).

As an initial approach to generalise the results across the time series prediction applications and noise types adopted in this study, we create Table 4.8. It captures the ranges of the parameter c (as an interval $[c_{left}, c_{right}]$) that have resulted in the

Table 4.8 Intervals with optimal FOU sizes (c) generalised across all time series and noise types in this thesis.

SNR	FOU size c interval $[c_{left}, c_{right}]$			
	c_{left}		c_{right}	
	minimum	average	maximum	average
NF	0	0	0	0
20	0.10	0.12	0.20	0.17
18	0.10	0.17	0.30	0.23
16	0.10	0.22	0.30	0.25
14	0.20	0.28	0.40	0.30
12	0.30	0.40	0.80	0.50
10	0.40	0.52	0.90	0.68
8	0.50	0.62	1.0	0.82
6	0.60	0.82	1.0	0.87
4	0.70	0.88	1.0	0.97
2	0.70	0.93	1.0	1.0
0	1.0	1.0	1.0	1.0

Table 4.9 Example computation for SNR=20 dB in table 4.8.

Table	c_{left}		c_{right}	
4.1	0.1		0.2	
4.2	0.1		0.2	
4.3	0.1		0.1	
4.4	0.1		0.1	
4.5	0.2		0.2	
4.6	0.1		0.2	
	minimum	average	maximum	average
	0.10	0.12	0.20	0.17

optimal performance for a given SNR across all experiments (Tables 4.1-4.6) and thus all time series and noise types. To generate this table, for each SNR level and for each experiment, the minimum and maximum c parameters are aggregated using either the mean or a min/max (for minimum and maximum c values respectively).

Table 4.9 provides a detailed example of the calculations underpinning Table 4.8. In it, we consider the results of SNR=20dB from Tables 4.1-4.6 and take the corresponding two c values that give the minimum RMSE (i.e., the bold values). The c value that results in the minimum RMSE value will be considered as c_{left} and the other one as c_{right}. After collecting all of these values from each table, the minimum and the average of c_{left} as well as the maximum and the average of c_{right} are calculated as shown in Table 4.9. The same method is applied to the other noise levels and their final results are summarised in Table 4.8.

Returning to the example of the SNR of 10db, the FOU size c for the 'general' time series prediction applications can be looked up from Table 4.8 to be in the interval [0.4,0.9] (based on min/max) or [0.52,0.68] (based on average).

4.6 Summary

In this chapter, an extensive series of Time-Series Prediction (TSP) experiments was conducted to investigate the relationships between the FOU size and uncertainty levels. In particular, we introduce an approach for the controlled blurring (CB) of type-1 MFs to interval type-2 MFs that ensures that the FOU size within a given IT2 fuzzy set remains constant (within the support of the lower membership function) and hence, satisfying requirements 1 and 2 in Chapter 3 (Section 3.2). The latter is crucial if we seek to establish a direct link between a known level of uncertainty affecting a given variable and the FOU size of the fuzzy sets describing that variable.

Using the controlled blurring method, CB, we investigated the relationship between the FOU size of T2 FSs and the level of the presented uncertainty. We demonstrated and analysed the performance of a range of carefully designed FLSs in the context of the three time series: Mackey Glass, Ikeda and Lorenz. A direct relationship between the FOU sizes of the FSs and the noise levels was shown and illustrated, namely, as noise level increases, the FOU that gives the minimum RMSE value increases as well. Also, it was observed that the T1 FLSs ($c = 0$) are outperforming IT2 FLSs with noise free (NF) testing data as is intuitively expected.

Based on the experimental results, we first provide specific examples of the optimal FOU size for specific TSP experiments before providing guidelines on FOU size selection based on aggregate results across all experiments conducted. While it was clear that these guidelines are based on TSP experiments, they provided a valuable starting point for establishing more generically applicable guidelines for FOU size selection in applications where the level of uncertainty (e.g., SNR in our case) can be determined a priori.

Finally, the proposed approach (CB) that enables the transition from T1 to IT2 FLSs through varying the size of the FOU of their respective FSs while keeping the size of the FOU over the FS as constant as possible was investigated. This is an important feature of this method as it enables the relating of FOU size to levels of uncertainty and vice versa. However, this method does not maintain the T1 MF shape, which starts for example with a triangular shape in T1 principal MFs, and as we increase the FOU size, the shape of IT2 FS changes to a different shape (i.e., trapezoidal).

Based on the results of following the systematic design and evaluation process employed in this study and the conclusions drawn, the first and second objectives stated in the introduction are met in this chapter. However, both of these objectives will be revisited again in Chapter 5 due to the shortcoming in CB stated above.

In the next chapter, we provide complete detail for the shape-preserving controlled blurring approach initially introduced in Section 3.2.3, followed by a series of in–depth experiments showing the performance of the resulting method.

Chapter 5

An Investigation into the Shape-Preserving Controlled Blurring for the Generation of Interval Type-2 Fuzzy Sets

5.1 Introduction

The objective of this chapter is to investigate the shape-preserving controlled blurring approach highlighted in Chapter 3, (SP-CB), which is an alternative FOU creation method to CB satisfying requirements 1, 2 and 3 in Section 3.2. The controlled blurring (CB) method has been thoroughly investigated in Chapter 4 and has a potential shortcoming in not preserving the initial T1 MFs shape as the FOU size increases. Keeping the original MF shape will enable an intuitive comparison between the T1 and T2 FSs.

This chapter builds on Chapter 4 and further explores the methodological transition of T1 to IT2 FSs for given 'levels' of uncertainty. Specifically, we detail and apply the shape-preserving controlled blurring (SP-CB) method, which proposes to transition from T1 to IT2 FLSs through varying the size of the FOU of their respective FSs while maintaining the original FS shape (e.g., triangular) and keeping the size of the FOU over the FS as constant as possible. The latter is important as it enables the systematic relating of FOU size to levels of uncertainty and vice versa, while the former enables an intuitive comparison between the T1 and T2 FSs. The effectiveness of the proposed method is demonstrated through a series of experiments using both the MG and LZ time series prediction problems. The results are compared with the results of the IT2 FS creation method (CB).

By using the two methods (CB and SP-CB), we will be able to investigate three issues not considered in Chapter 4 that highlight the potential advantages and disadvantages of both approaches in theoretical work and practical applications. Specifically, Chapter 5 comprises the following:

- The FOU creation methods (CB and SP-CB) and their comparison.
- Training and testing the FLSs using an NF data set and noisy data set. Training and testing using the same noise level (i.e. SNR) is the case in most real world cases, but it is also possible that a system is trained using NF data, for example as in laboratory conditions, can then be used in real world conditions (noisy). So, we apply our experiments using both NF and noisy data.
- Using both singleton and non-singleton fuzzification. As the SNR gets smaller (more uncertainty), non-singleton fuzzification becomes more important and most likely will improve the system's performance.

In the remaining sections of this chapter, we will detail the shape-preserving controlled blurring (SP-CB) method (in Section 5.2). A series of experiments that illustrate the FOU construction approaches and follow the IT2 FS design and evaluation process highlighted in Chapter 3 (Section 3.3) will be detailed in section 5.3 and there we will present their results. The results will then be analysed and discussed in section 5.4, followed by the chapter summary in section 5.5.

5.2 The Shape-Preserving Controlled Blurring Method Details

In an effort to find an alternative FOU creation method satisfying requirements 1, 2 and 3 presented in Section 3.2, to address the shortcoming of CB method which starts for example with a triangular shape in T1 principal MFs and, as we increase the FOU size, the shape of IT2 FS changes to a different shape, a new method that enables a shape-preserving is proposed. This shape-preserving controlled blurring (SP-CB) method is used to obtain an IT2 FS with a uniform FOU that captures a constant level of uncertainty over the core of the fuzzy set based on the bounded values of uncertainty δ and a fixed parameter $c \in [0,1]$. The parameters are used to create an FOU of a given size around a principal T1 MF and to maintain the same shape of the T1 MF for both the UMF and the LMF as the FOU increases. We will refer to this approach as SP-CB method throughout the thesis.

We start the design of the IT2 FSs by including the FOU size parameter c and the bounded values of uncertainty δ to form the IT2 FSs of the system. Note that when both c and δ are equal to zero, the result is the original T1 FS MF, while $c = 1$ results in an IT2 set with a very wide FOU (as detailed below).

In order to preserve the original shape of the membership function (T1 MF) as we add a certain value of the FOU size parameter c to it to create an IT2 FS, we construct the UMF, $\overline{\mu}(x)$, and the LMF, $\underline{\mu}(x)$, of the IT2 FS by adding/subtracting the specific value of (δ) from the T1 triangular and trapezoidal MFs parameters as follows:

$$\underline{a} = a + \delta \tag{5.1a}$$

$$\overline{a} = a - \delta \tag{5.1b}$$

$$\underline{b} = b - \delta \tag{5.1c}$$

$$\overline{b} = b + \delta \tag{5.1d}$$

where $\underline{a}$ and $\underline{b}$ are the left and the right points of the LMFs, $\overline{a}$ and $\overline{b}$ are the left and the right points of the UMFs.

In the case of Gaussian MF, we construct the UMF, $\overline{\mu}(x)$, and the LMF, $\underline{\mu}(x)$, of the IT2 FS by adding/subtracting the specific value of (δ) from the T1 Gaussian MF parameters as follows:

$$\overline{\sigma} = \sigma + \delta \tag{5.2a}$$

$$\underline{\sigma} = \sigma - \delta \tag{5.2b}$$

where $\overline{\sigma}$ and $\underline{\sigma}$ are the standard deviation of the UMF and LMF respectively.

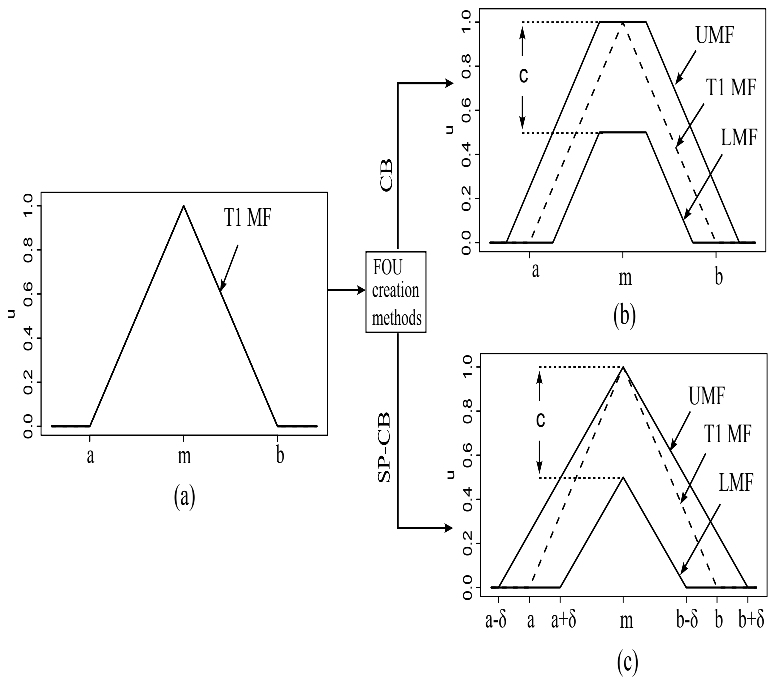


Figure 5.1 Illustration of the design methods of triangular IT2 MFs with the FOU size parameter $c = 0.5$. (a) Initial T1 FS, (b) IT2 FS created using CB method and (c) IT2 FS created using SP-CB method.

The UMF, $\overline{\mu}(x)$ and the LMF, $\underline{\mu}(x)$ of IT2 FS for triangular case can then be obtained as follows:

$$\overline{\mu}(x) = \begin{cases} 0 & \overline{b} \leq x \leq \overline{a} \\ \frac{x-\overline{a}}{m-\overline{a}} & \overline{a} \leq x \leq m \\ \frac{\overline{b}-x}{\overline{b}-m} & m \leq x \leq \overline{b} \end{cases} \tag{5.3}$$

$$\underline{\mu}(x) = (1-c) * \begin{cases} 0 & \underline{b} \leq x \leq \underline{a} \\ \frac{x-\underline{a}}{m-\underline{a}} & \underline{a} \leq x \leq m \\ \frac{\underline{b}-x}{\underline{b}-m} & m \leq x \leq \underline{b} \end{cases} \tag{5.4}$$

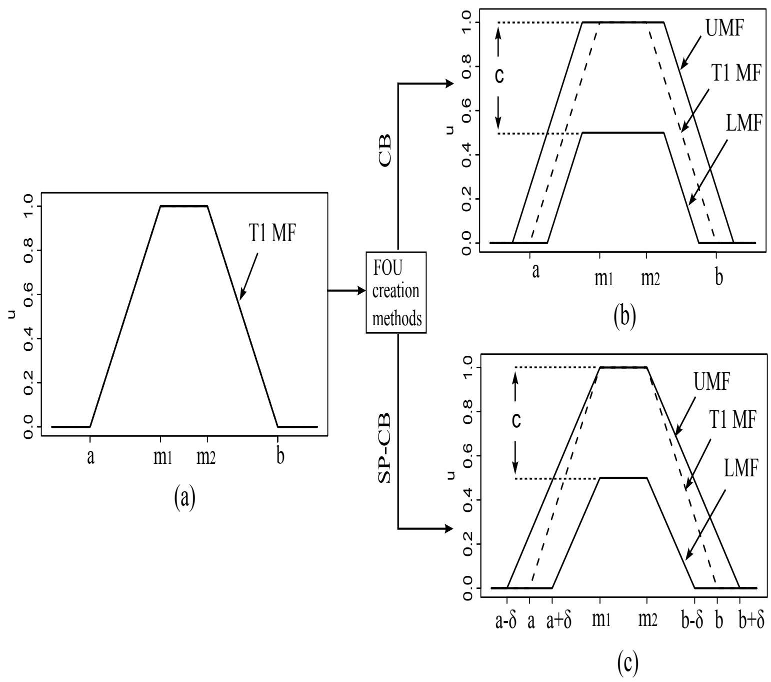


Figure 5.2 Illustration of the design methods of trapezoidal IT2 MFs with the FOU size parameter $c = 0.5$. (a) Initial T1 FS, (b) IT2 FS created using CB method and (c) IT2 FS created using SP-CB method.

For the trapezoidal case, the UMF, $\overline{\mu}(x)$, and the LMF, $\underline{\mu}(x)$, of IT2 FS can then be obtained as follows:

$$\overline{\mu}(x) \;=\; \begin{cases} 0 & \overline{b} \leq x \leq \overline{a} \\ \frac{x-\overline{a}}{m_1-\overline{a}} & \overline{a} \leq x \leq m_1 \\ 1 & m_1 \leq x \leq m_2 \\ \frac{\overline{b}-x}{\overline{b}-m_2} & m_2 \leq x \leq \overline{b} \end{cases} \tag{5.5}$$

$$\underline{\mu}(x) \;= (1-c) * \begin{cases} 0 & \underline{b} \leq x \leq \underline{a} \\ \frac{x-\underline{a}}{m_1-\underline{a}} & \underline{a} \leq x \leq m_1 \\ 1 & m_1 \leq x \leq m_2 \\ \frac{\underline{b}-x}{\underline{b}-m_2} & m_2 \leq x \leq \underline{b} \end{cases} \tag{5.6}$$

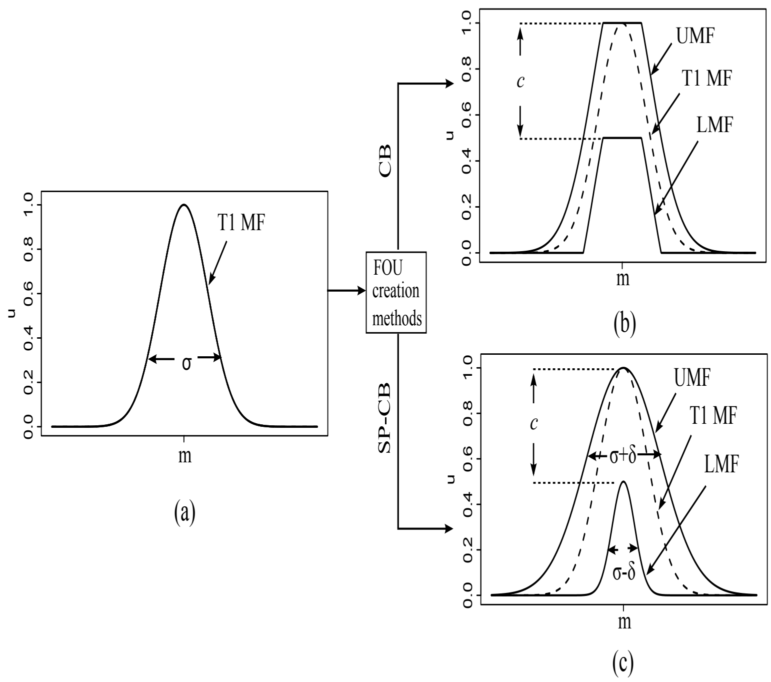


Figure 5.3 Illustration of the design methods of Gaussian IT2 MFs with the FOU size parameter $c = 0.5$. (a) Initial T1 FS, (b) IT2 FS created using CB method and (c) IT2 FS created using SP-CB method.

Whereas, for the Gaussian case, the UMF, $\overline{\mu}(x)$, and the LMF, $\underline{\mu}(x)$, of IT2 FS can then be obtained as follows:

$$\overline{\mu}(x) \;= e^{\frac{-(x-m)^2}{2\overline{\sigma}^2}} \tag{5.7}$$

$$\underline{\mu}(x) \;= (1-c) * e^{\frac{-(x-m)^2}{2\underline{\sigma}^2}} \tag{5.8}$$

In (5.4), (5.6) and (5.8) we scaled the lower membership function by $(1-c)$ to create a uniform FOU all over the IT2 FS. Parameter $c \in [0,1]$ is the FOU size parameter. The δ parameter is used to define the endpoints in the fuzzy MFs (see Figure 5.1, Figure 5.2 and Figure 5.3). Its value can be obtained experimentally, so that, for a given value of c, the uncertainty indicator, $U_{\tilde{A}}(x)$ in (3.1), is constant (equal to c) over the support of the LMF ($\underline{\mu}(x) > 0$).

A more detailed illustration of the design of the IT2 FS is depicted in Figure 5.1(a) and 5.1(c) for triangular MFs, Figure 5.2(a) and 5.2(c) for trapezoidal MFs and Figure 5.3(a) and 5.3(c) for Gaussian MFs using SP-CB method.

After considering the proposed FOU creation techniques, we proceed to an experimental exploration of the different behaviour of IT2 FSs created using either the CB or the SP-CB methods when transitioning from a T1 to an IT2 FLSs.

5.3 Experiments and Results

FLSs have been successfully used in forecasting of time series [8–10],[13]. As the level of uncertainty/noise is controllable, we use time series prediction here as a test bed to explore the different approaches to IT2 FLS generation. We conduct a series of experiments using both the Mackey-Glass (MG) and Lorenz (LZ) time series, a chaotic time series; their details were given in in Chapter 4 Sections (4.3.1) and (4.3.3) respectively.

Following Chapter 3 (Section 3.3) that provides a general design process framework. It is designed to systematically describe the initial design of the T1 FLS for a given application and its subsequent transformation to one or more IT2 FLSs by applying the FOU creation methods. Each of the process steps is detailed below and applied to in the same manner to the MG and LZ time series, with the results presented. For each, we show three cases based on different type of FSs: triangular, trapezoidal and Gaussian MFs.

First, we generate a data set (both training and testing data) from each of the two time series. Then, we design evenly distributed T1 FSs and create the rule base by applying the WM approach using the training data sets: noise-free training (NFtrain) set and training sets corrupted by different noise levels (Ntrain).

Next, we start the design of a series of IT2 FLSs by generating IT2 FSs using the FOU size parameter c to form the IT2 MFs of the systems. The actual number of FSs and the rules from the T1 system are maintained. In parallel, different levels of noise as a source of uncertainty are employed to generate a series of testing data sets. Finally, the performance of each of the T1/IT2 FLSs is evaluated for each of the testing data sets and a comparison is made between the CB and the SP-CB methods.

The complete process is illustrated by the flowchart in Figure 5.4 and can be summarised in the four subsections (5.3.1 - 5.3.4) below.

5.3.1 Data Generation

Following Chapter 3 (Section 3.3.1), NF data for the two time series are generated using (4.9) for the MG and (4.11a, 4.11b, 4.11c) for the LZ time series with the parameters and the numerical solutions of the differential equations presented in Chapter 4, Section (4.3.1 and (4.3.3) respectively. For the LZ, we use variable x in this study.

To obtain training and testing input-output data pairs, we specify the value of $N = 700$, $D = 500$ and $p = 4$. Specifically, we extract 700 input-output data pairs. In our case, four-input, one-output FLSs (single stage prediction) are considered. The training data set is the first 500 data points used for training the FLSs (generating the rules) using $x(1001)$ to $x(1504)$. The following 200 points using $x(1505)$ to $x(1708)$

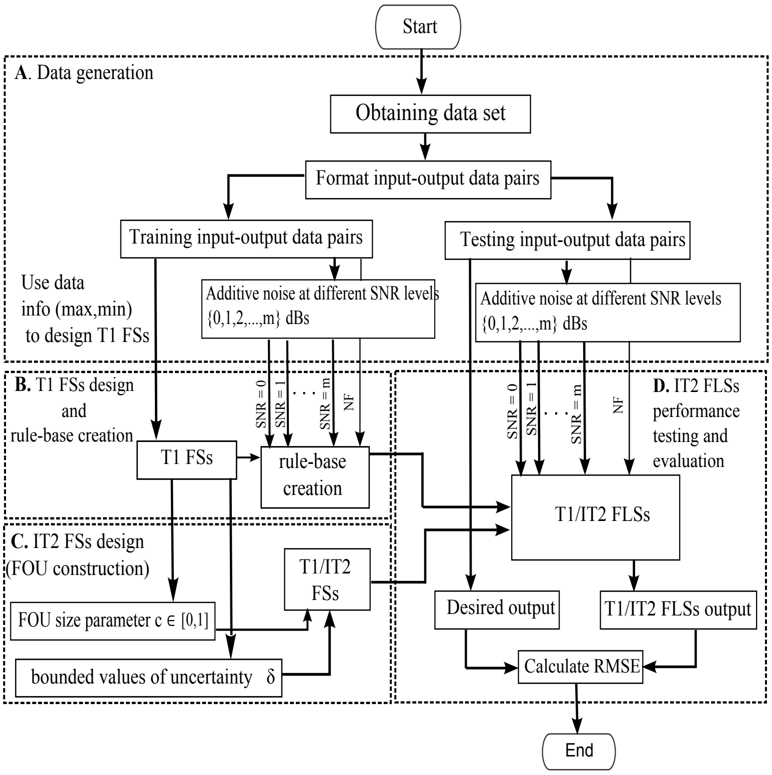


Figure 5.4 A flowchart of the process of using different FOU sizes to design and evaluate IT2 FLSs at different noise levels for the SP-CB method.

are used for testing the FLSs. In this study, we consider different versions of training and testing data corrupted with zero-mean uniform noise for different SNRs. We use 5 noise levels in training and the same number in testing. The chosen noise levels are (from lowest to highest level): 20,16,10,4 and 0 dBs. The original noise free (NF) data also used for training and testing the FLSs.

The training data sets used in our experiments include:

- One set of 500 NF input-output training pairs. This is denoted noise-free training (NFtrain) set.

- Five sets of 500 noisy input-output training pairs corrupted with zero-mean uniform noise for five different SNR values: 0, 4, 10, 16 and 20 dBs. These are denoted noisy training (Ntrain) data sets.

For the testing data, we generate:

- One set of 200 NF input-output testing pairs. This is denoted noise-free testing (NFtest) set.
- Five sets of 200 noisy input-output testing pairs corrupted with zero-mean uniform noise for five different (noise levels) SNR values: 0, 4, 10, 16 and 20 dBs.

5.3.2 Type-1 Fuzzy Sets Design and Rule-Base Creation

Following Chapter 3 (Section 3.3.2) and these experiment, we choose the number of MFs (triangular, trapezoidal or Gaussian) to be 7 for each input and output of the FLS. These are defined evenly to cover the input and output spaces. The triangular, trapezoidal and Gaussian T1 MFs used for the inputs and outputs are shown in Figure 5.5(a), Figure 5.6(a) and Figure 5.7(a) respectively. The MFs are labelled using numbers (e.g., F11 represents FS 1 of input 1 and for the output F1 represent FS 1). Then, we apply the well-known WM method to create the rules for the given training input-output dataset that are corrupted with different levels of noise. In this work we used 5 noise levels in training which include 20,16,10,4 and 0 dBs. The original NF data was also used for training the FLSs. The result is 18 rule bases (6 for triangular MF, 6 for trapezoidal MF and 6 for Gaussian MF) generated using different noise levels as shown in Table 5.1. The actual number of FSs and the rules are maintained

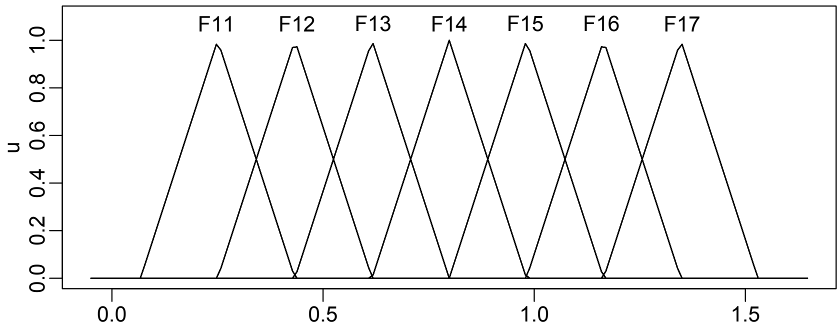
Table 5.1 Number of rules generated at each noise level using the WM-method applied to T1 FLSs using triangular, trapezoidal and Gaussian MFs with MG and LZ.

Noise Level	MFs with MG time series			MFs with LZ time series		
	triangular	trapezoidal	Gaussian	triangular	trapezoidal	Gaussian
NF	47	47	47	70	70	70
20	80	80	80	91	91	91
16	91	91	91	98	98	98
10	125	125	125	142	142	142
4	283	283	283	260	260	260
0	330	330	330	307	307	307

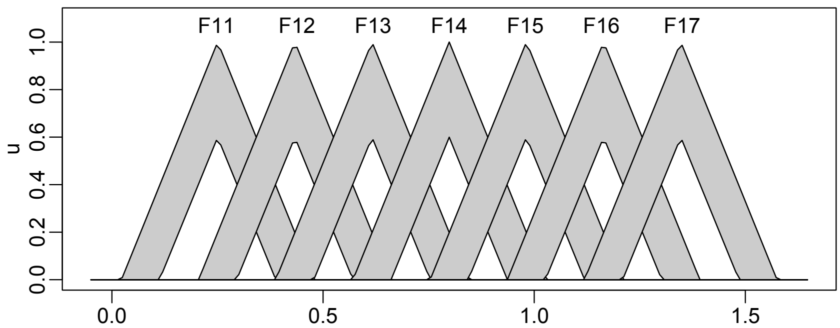
from the T1 FLSs and used for all resulting IT2 FLSs. From Table 5.1, we can see that triangular, trapezoidal and Gaussian MFs used in T1 FSs generate the same number of rules at each noise level for each of MG and LZ time series.

5.3.3 Interval Type-2 Fuzzy Sets Creation

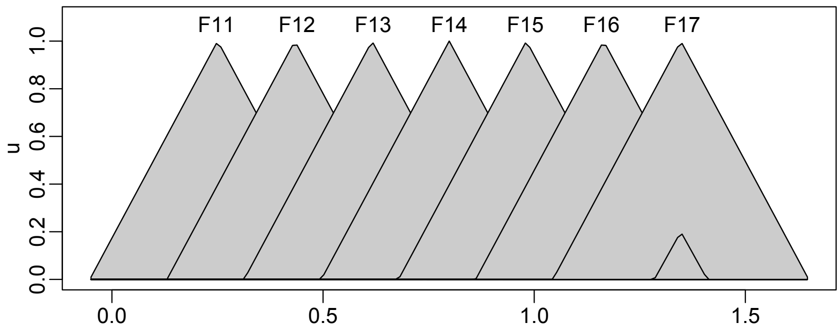
Following Chapter 3 (Section 3.3.3), we extend the T1 FLSs into a series of IT2 FLSs. First, we design the T1 FSs for triangular, trapezoidal and Gaussian MFs and create the rule bases employing the WM approach using the training data set. Then, we start the design of a series of IT2 FLSs by generating IT2 FSs using the FOU creation methods CB and SP-CB to form the IT2 MFs of the systems. The actual number of FSs and the rules are maintained from the T1 system. All the common parameters of the singleton fuzzy logic systems (SFLSs) and non-singleton fuzzy logic systems (NSFLSs) are the same. For all experiments, σ_x in the NSFLS case is set equal to the



(a)

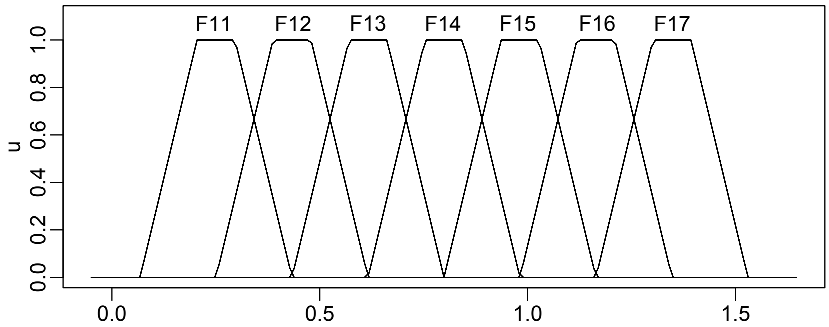


(b)

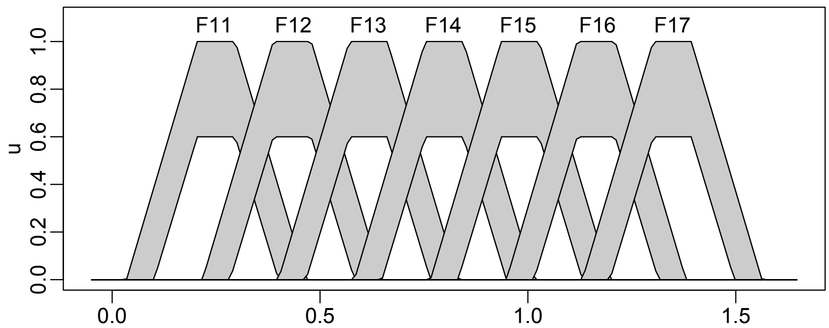


(c)

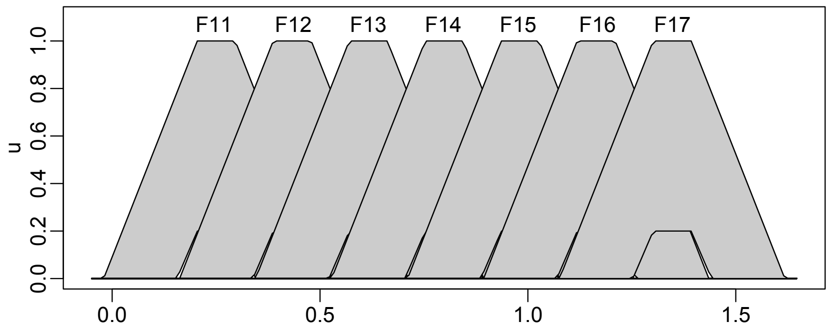
Figure 5.5 Triangular T1 and two examples of IT2 MFs used in the design of the TSP FLSs using the SP-CB method. (a) T1 MFs, (b) IT2 MFs designed with FOU size parameter $c = 0.4$ and (c) IT2 MFs designed with FOU size parameter $c = 0.8$.



(a)

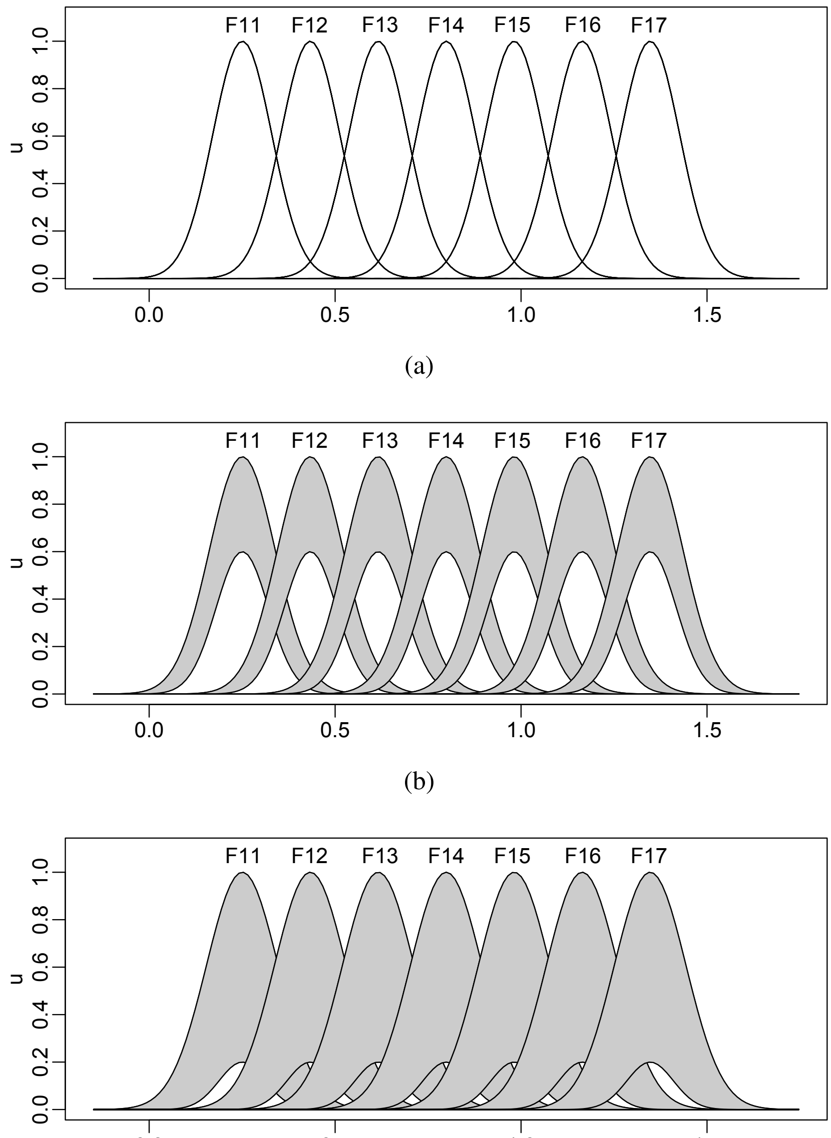


(b)



(c)

Figure 5.6 Trapezoidal T1 and two examples of IT2 MFs used in the design of the TSP FLSs using the SP-CB method. (a) T1 MFs, (b) IT2 MFs designed with FOU size parameter $c = 0.4$ and (c) IT2 MFs designed with FOU size parameter $c = 0.8$.



(a)

(b)

(c)

Figure 5.7 Gaussian T1 and two examples of IT2 MFs used in the design of the TSP FLSs using the SP-CB method. (a) T1 MFs, (b) IT2 MFs designed with FOU size parameter $c = 0.3$ and (c) IT2 MFs designed with FOU size parameter $c = 0.8$.

standard deviation of the additive noise. In a noise free situation for which $\sigma_x = 0$, the performance of the NSFLS is identical to that of the SFLS.

To construct the upper and lower membership functions of the IT2 FSs, we use the following:

- The FOU parameter c values used for both methods (CB and SP-CB) are chosen to be 0,0.2,0.4,0.8 and 1.0. At $c = 0$ the IT2 FSs reduce to the original T1 FSs, whereas in case of using $c = 1$, the IT2 FSs reach the maximum amount of their width.

- For the CB method, we use the method detailed in Chapter 4 and create the IT2 FSs by applying the equations (4.1) and (4.2) for triangular, trapezoidal and Gaussian cases.

- For the SP-CB method, we use the method detailed in this chapter and design the IT2 FSs by applying the equations (5.3) and (5.4) for triangular case, the equations (5.5) and (5.6) for trapezoidal case and the equations (5.7) and (5.8) for Gaussian case.

- For the SP-CB, the δ parameter is used to define the bounded values of the MFs and its values can be obtained experimentally, so that, for a given value of c parameter and using the uncertainty indicator (UI), $U_{\tilde{A}}(x)$ in (3.1), the UI value should be constant within the supports of the lower MFs ($\underline{\mu}_{\tilde{A}}(x) > 0$) with a value equal to c. δ values for triangular, trapezoidal and Gaussian FSs based on both MG and LZ time series are shown in Table 5.2.

Examples of the FOU construction using the SP-CB method for the FSs of two IT2 FLSs using different FOU size parameters $c = 0.40$ and $c = 0.80$ are shown in

Table 5.2 Values of δ for triangular, trapezoidal and Gaussian FSs with MG and LZ.

c value	δ value with MG time series			δ value with LZ time series		
	triangular	trapezoidal	Gaussian	triangular	trapezoidal	Gaussian
0	0	0	0	0	0	0
0.2	0.02	0.015	0.004	0.60	0.45	0.191
0.4	0.045	0.033	0.009	1.35	1.02	0.40
0.8	0.12	0.088	0.017	3.60	2.70	1.10
1.0	0.182	0.127	0.021	5.40	2.77	1.72

Figure 5.5(b) and Figure 5.5(c) for the triangular case, examples of the trapezoidal case are shown in Figure 5.6(b) and Figure 5.6(c) and examples of the Gaussian case are shown in Figure 5.7(b) and Figure 5.7(c).

Table 5.3 shows the triangular, trapezoidal and Gaussian FSs at different c values for the CB method and SP-CB method. Note how both methods maintain a constant level of uncertainty (FOU size) over the support of the LMFs, while the SP-CB method also preserves the shape of the T1 MFs.

After creating the IT2 FSs (triangular, trapezoidal and Gaussian) for both CB and SP-CB, we start to design SFLSs and NSFLSs using the rule bases generated previously as described in Section 5.3.2.

After finishing the design of the IT2 FLSs with the chosen FOU sizes for both methods at different configurations as mentioned above, the testing data sets are used to test the performance of the individual IT2 FLSs when faced with different uncertainty/noise levels. Now we have a total of 20 FLSs for each design method with

Table 5.3 Triangular,trapezoidal and Gaussian FSs at different c values for the CB and the SP-CB methods.

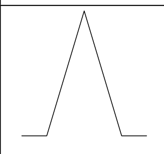
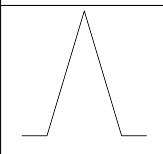
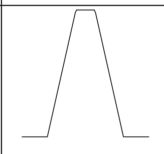
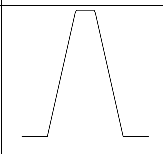
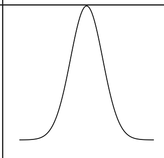
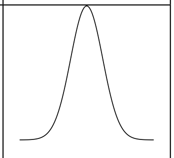
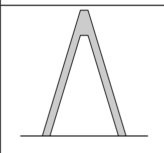
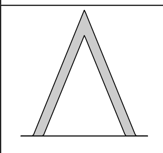
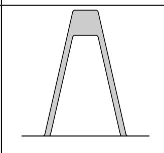
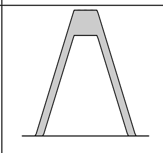
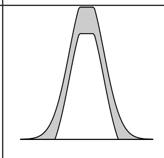
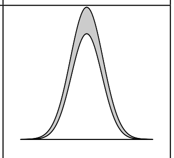
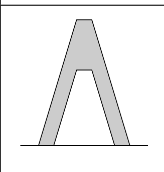
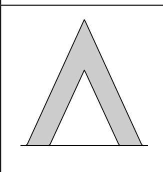
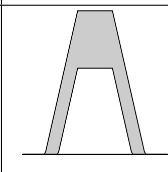
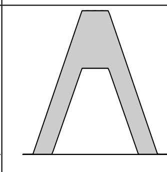
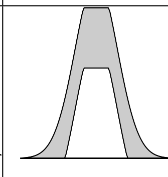
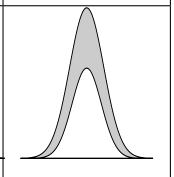
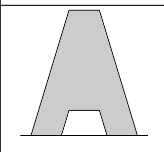
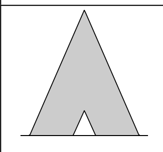
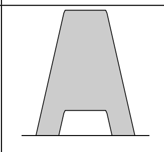
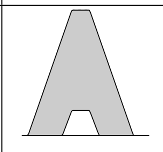
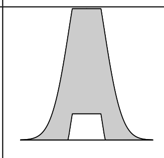
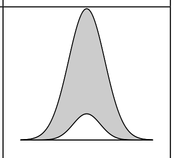
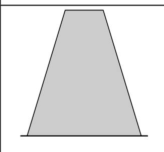
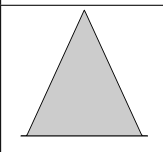
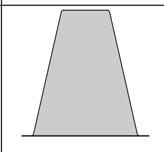
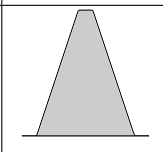
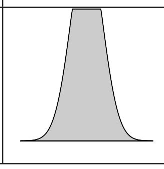
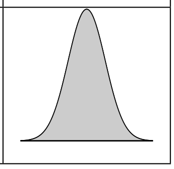
	triangular		trapezoidal		Gaussian	
c	CB	SP-CB	CB	SP-CB	CB	SP-CB
0						
0.2						
0.4						
0.8						
1.0						

Table 5.4 The general structure of the experiments for each of MG and LZ.

No. of FLSs	Method	MF Type	Training Type	Fuzzification Type	FOU sizes
5	CB	triangular	NFtrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	CB	triangular	NFtrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	CB	triangular	Ntrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	CB	triangular	Ntrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	CB	trapezoidal	NFtrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	CB	trapezoidal	NFtrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	CB	trapezoidal	Ntrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	CB	trapezoidal	Ntrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	CB	Gaussian	NFtrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	CB	Gaussian	NFtrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	CB	Gaussian	Ntrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	CB	Gaussian	Ntrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	triangular	NFtrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	triangular	NFtrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	triangular	Ntrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	triangular	Ntrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	trapezoidal	NFtrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	trapezoidal	NFtrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	trapezoidal	Ntrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	trapezoidal	Ntrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	Gaussian	NFtrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	Gaussian	NFtrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	Gaussian	Ntrain	Singleton	{0,0.2,0.4,0.8,1.0}
5	SP-CB	Gaussian	Ntrain	Non-singleton	{0,0.2,0.4,0.8,1.0}
120 FLSs	Total				
Testing	{NFtest,20,16,10,4,0}				

each MF case (triangular, trapezoidal and Gaussian) (i.e. 5 levels of c * 2 training conditions (NFtrain and Ntrain) * 2 FLS types (NSFLS, SFLS)). The total number of constructed FLSs for each time series is 120 FLSs (20 with triangular (CB method), 20 with triangular (SP-CB method) 20 for trapezoidal (CB method) and 20 for trapezoidal (SP-CB) 20 with Gaussian (CB method), 20 with Gaussian (SP-CB method)) and each system will be tested over 5 different noise levels and the NFtest case. The general structure of the experiments is depicted in Table 5.4. The testing and evaluation of these systems will be presented next.

5.3.4 Fuzzy Logic System Evaluation

We test each of the IT2 FLSs described above against 6 levels of noise in order to determine the best performance for each noise level. Each test is repeated 30 times to account for the random generation of the uniform noise. The performances of all the designs were evaluated using their RMSE (5.9) over the testing data set.

$$RMSE = \sqrt{\frac{1}{200}\sum_{t=1508}^{1707}\left[s\left(t+1\right)-f\left(\mathbf{s}^{(t)}\right)\right]^2} \tag{5.9}$$

where, $s(t+1)$ is the output of the noisy testing data and $f(s^{(t)})$ is the crisp output of the FLS, and, $\mathbf{s}^{(t)} = [s(t-3), s(t-2), s(t-1), s(t)]^T$.

In addition, percentage of improvement based on RMSE (PI_{RMSE}) [172], for example, obtained through the CB method over SP-CB method can be calculated using (5.10) shown below.

$$PI_{RMSE} = \frac{(SP-CB)-(CB)}{CB} \times 100\% \tag{5.10}$$

The RMSE results are averaged over 30 runs and are depicted in Tables 5.5-5.10 showing the results of the average RMSE of each of the MG and LZ time series using three different MF types (triangular, trapezoidal and Gaussian) with the two different design methods (CB and SP-CB). Each column represents an IT2 FLS design with a given FOU size parameter c. The rows show the average RMSE value at the different SNR values for all FOU sizes/FLSs. The shaded values are the result of the SP-CB method.

5.4 Analysis and Discussion

In order to explore the viability and effects of the FOU creation method SP-CB, in particular in relation to the CB method, we briefly discuss the results of the experiments conducted in the previous sections. Illustrations of sample outputs for MG and LZ TSP resulting from applying the SP-CB method are found in Figures 5.8 and 5.9 respectively.

After analysing of the RMSE results of both methods, we can divide the analysis into three areas. The first analysis covers a comparison of the FLSs based on CB and SP-CB in terms of their performance in general. The second analysis is devoted to the performance of CB and SP-CB in different settings (i.e., training on NFtrain and Ntrain data, SFLSs and NSFLSs). The third analysis includes the performance analysis of the FLSs with the three cases (triangular, trapezoidal and Gaussian MFs) on different settings for both MG and LZ time series. Finally, common observations are presented.

Table 5.5 The average RMSE values for MG time series prediction for triangular MFs (shaded values are the results of SP-CB method).

	c=0				c=0.2				c=0.4				c=0.8				c=1.0			
	NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain	
	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS
NFtest	0.0313	0.0313	0.0313	0.0313	0.0318	0.0318	0.0318	0.0318	0.0369	0.0369	0.0369	0.0369	0.0593	0.0593	0.0593	0.0593	0.0784	0.0784	0.0784	0.0784
	0.0313	0.0313	0.0313	0.0313	0.0321	0.0321	0.0321	0.0321	0.0400	0.0400	0.0400	0.0400	0.0679	0.0679	0.0679	0.0679	0.0914	0.0914	0.0914	0.0914
20	0.0426	0.0425	0.0540	0.0539	0.0423	0.0424	0.0538	0.0538	0.0451	0.0451	0.0553	0.0553	0.0610	0.0621	0.0636	0.0642	0.0765	0.0752	0.0771	0.0775
	0.0426	0.0425	0.0540	0.0539	0.0424	0.0425	0.0539	0.0539	0.0468	0.0466	0.0562	0.0561	0.0741	0.0735	0.0766	0.0759	0.0897	0.0903	0.0903	0.0921
16	0.0554	0.0551	0.0697	0.0690	0.0549	0.0548	0.0686	0.0680	0.0563	0.0558	0.0683	0.0680	0.0671	0.0669	0.0802	0.0797	0.0800	0.0783	0.0977	0.0960
	0.0554	0.0551	0.0697	0.0690	0.0550	0.0548	0.0685	0.0680	0.0570	0.0562	0.0690	0.0686	0.0773	0.0755	0.0955	0.0903	0.0877	0.0901	0.0958	0.1018
10	0.0997	0.0918	0.1039	0.1006	0.0927	0.0904	0.0996	0.0967	0.0917	0.0875	0.0914	0.0898	0.0923	0.0831	0.0840	0.0923	0.1024	0.1004	0.1087	0.1125
	0.0997	0.0918	0.1039	0.1006	0.0929	0.0904	0.0995	0.0967	0.0919	0.0870	0.0908	0.0894	0.0985	0.0867	0.1055	0.0988	0.1000	0.1026	0.1112	0.1175
4	0.3355	0.1526	0.1971	0.1747	0.2639	0.1488	0.1832	0.1664	0.1934	0.1401	0.1603	0.1511	0.1430	0.1239	0.1454	0.1038	0.1487	0.1399	0.1498	0.1611
	0.3355	0.1526	0.1971	0.1747	0.2816	0.1494	0.1851	0.1674	0.2135	0.1405	0.1613	0.1512	0.1429	0.1234	0.1467	0.1393	0.1444	0.1359	0.1498	0.1655
0	0.5150	0.2088	0.3023	0.2519	0.4418	0.2039	0.2797	0.2410	0.3714	0.1928	0.2423	0.2191	0.2632	0.1692	0.2024	0.1839	0.2275	0.1819	0.2010	0.1947
	0.5150	0.2088	0.3023	0.2519	0.4691	0.2052	0.2854	0.2429	0.4127	0.1948	0.2506	0.2212	0.2789	0.1697	0.2038	0.1835	0.2129	0.1817	0.1991	0.1968

Table 5.6 The average RMSE values for MG time series prediction for trapezoidal MFs (shaded values are the results of SP-CB method).

	c=0				c=0.2				c=0.4				c=0.8				c=1.0			
	NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain	
	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS
NFtest	0.0312	0.0312	0.0312	0.0312	0.0316	0.0316	0.0316	0.0316	0.0370	0.0370	0.0370	0.0370	0.0581	0.0581	0.0581	0.0581	0.0657	0.0657	0.0657	0.0657
	0.0312	0.0312	0.0312	0.0312	0.0316	0.0316	0.0316	0.0316	0.0366	0.0366	0.0366	0.0366	0.0710	0.0710	0.0710	0.0710	0.0739	0.0739	0.0739	0.0739
20	0.0424	0.0424	0.0535	0.0534	0.0422	0.0423	0.0531	0.0530	0.0457	0.0457	0.0546	0.0545	0.0600	0.0595	0.0621	0.0623	0.0678	0.0757	0.0701	0.0753
	0.0424	0.0424	0.0535	0.0534	0.0422	0.0423	0.0531	0.0530	0.0449	0.0449	0.0539	0.0538	0.0659	0.0662	0.0679	0.0679	0.0741	0.0771	0.0765	0.0806
16	0.0553	0.0550	0.0715	0.0706	0.0550	0.0548	0.0703	0.0695	0.0572	0.0566	0.0696	0.0691	0.0665	0.0652	0.0762	0.0757	0.0737	0.0792	0.0875	0.0960
	0.0553	0.0550	0.0715	0.0706	0.0550	0.0548	0.0705	0.0697	0.0564	0.0559	0.0693	0.0688	0.0698	0.0686	0.0778	0.0771	0.0782	0.0829	0.0942	0.0945
10	0.0995	0.0914	0.1059	0.1027	0.0929	0.0902	0.1026	0.0996	0.0923	0.0884	0.0966	0.0950	0.0936	0.0848	0.1009	0.0995	0.0990	0.0981	0.1093	0.1167
	0.0995	0.0914	0.1059	0.1027	0.0931	0.0905	0.1034	0.1005	0.0921	0.0883	0.0973	0.0957	0.0923	0.0830	0.1004	0.0992	0.0998	0.1045	0.1153	0.1215
4	0.3352	0.1513	0.2030	0.1796	0.2809	0.1481	0.1915	0.1724	0.2231	0.1409	0.1727	0.1615	0.1603	0.1264	0.1593	0.1550	0.1492	0.1431	0.1615	0.1648
	0.3352	0.1513	0.2030	0.1796	0.2922	0.1492	0.1948	0.1753	0.2486	0.1425	0.1767	0.1648	0.1541	0.1253	0.1588	0.1548	0.1468	0.1394	0.1626	0.1665
0	0.5149	0.2072	0.2950	0.2373	0.4590	0.2029	0.2738	0.2257	0.4033	0.1934	0.2380	0.2072	0.3105	0.1719	0.1998	0.1814	0.2750	0.1830	0.1987	0.1898
	0.5149	0.2072	0.2950	0.2373	0.4803	0.2046	0.2807	0.2305	0.4407	0.1964	0.2492	0.2136	0.3310	0.1728	0.2013	0.1824	0.2720	0.1813	0.1984	0.1910

Table 5.7 The average RMSE values for MG time series prediction for Gaussian MFs (shaded values are the results of SP-CB method).

	c=0				c=0.2				c=0.4				c=0.8				c=1.0			
	NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain	
	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS
NFtest	0.0332	0.0332	0.0332	0.0332	0.0344	0.0344	0.0344	0.0344	0.0397	0.0397	0.0397	0.0397	0.0600	0.0600	0.0600	0.0600	0.0754	0.0754	0.0754	0.0754
	0.0332	0.0332	0.0332	0.0332	0.0339	0.0339	0.0339	0.0339	0.0373	0.0373	0.0373	0.0373	0.0542	0.0542	0.0542	0.0542	0.0632	0.0632	0.0632	0.0632
20	0.0424	0.0425	0.0536	0.0537	0.0428	0.0430	0.0540	0.0541	0.0458	0.0461	0.0560	0.0562	0.0616	0.0618	0.0649	0.0652	0.0742	0.0740	0.0749	0.0747
	0.0424	0.0425	0.0536	0.0537	0.0426	0.0428	0.0538	0.0539	0.0443	0.0444	0.0550	0.0551	0.0566	0.0568	0.0621	0.0622	0.0650	0.0650	0.0669	0.0669
16	0.0538	0.0537	0.0632	0.0632	0.0539	0.0539	0.0636	0.0636	0.0549	0.0551	0.0662	0.0666	0.0646	0.0650	0.0773	0.0778	0.0747	0.0748	0.0834	0.0834
	0.0538	0.0537	0.0633	0.0632	0.0538	0.0538	0.0631	0.0634	0.0543	0.0543	0.0647	0.0648	0.0605	0.0608	0.0718	0.0721	0.0672	0.0676	0.0775	0.0778
10	0.0890	0.0880	0.0942	0.0929	0.0882	0.0870	0.0923	0.0911	0.0852	0.0845	0.0899	0.0897	0.0795	0.0801	0.0939	0.0949	0.0830	0.0833	0.0988	0.0992
	0.0890	0.0880	0.0942	0.0929	0.0884	0.0874	0.0931	0.0920	0.0864	0.0854	0.0907	0.0902	0.0782	0.0780	0.0907	0.0911	0.0791	0.0794	0.0937	0.0942
4	0.1517	0.1462	0.1695	0.1591	0.1480	0.1425	0.1624	0.1526	0.1390	0.1353	0.1489	0.1439	0.1196	0.1182	0.1393	0.1395	0.1147	0.1128	0.1400	0.1415
	0.1517	0.1462	0.1695	0.1591	0.1499	0.1445	0.1653	0.1562	0.1443	0.1391	0.1544	0.1486	0.1265	0.1221	0.1405	0.1394	0.1205	0.1165	0.1391	0.1383
0	0.2130	0.2022	0.2539	0.2293	0.2043	0.1969	0.2428	0.2195	0.1915	0.1874	0.2188	0.2048	0.1678	0.1647	0.1883	0.1853	0.1552	0.1484	0.1837	0.1833
	0.2130	0.2022	0.2539	0.2293	0.2107	0.2001	0.2470	0.2251	0.2037	0.1934	0.2295	0.2134	0.1819	0.1722	0.1957	0.1893	0.1752	0.1646	0.1887	0.1847

Table 5.8 The average RMSE values for LZ time series prediction for triangular MFs (shaded values are the results of SP-CB method).

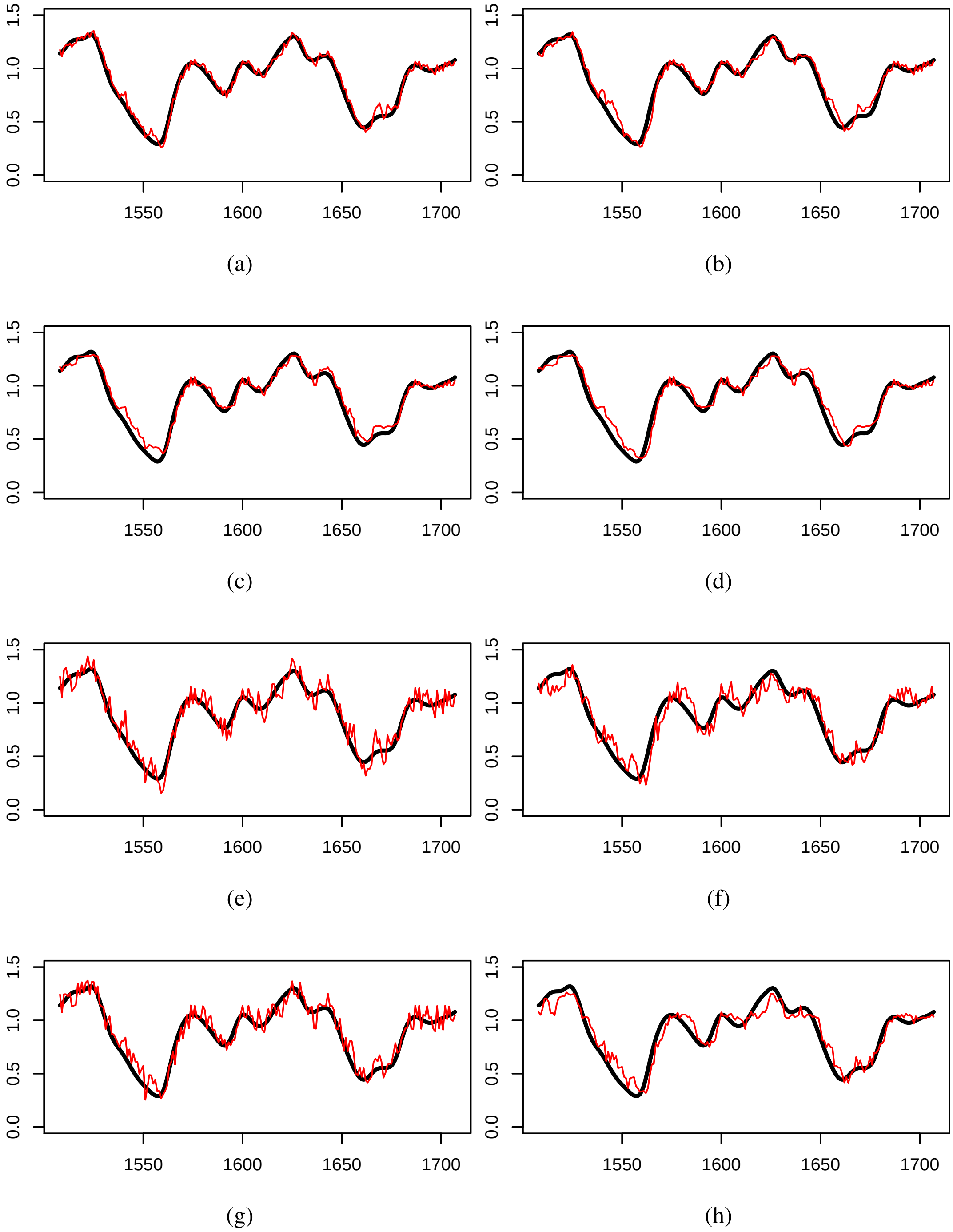
	c=0				c=0.2				c=0.4				c=0.8				c=1.0			
	NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain	
	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS
NFtest	1.9038	1.9038	1.9038	1.9038	1.9209	1.9209	1.9209	1.9209	2.0246	2.0246	2.0246	2.0246	2.9614	2.9614	2.9614	2.9614	3.7637	3.7637	3.7637	3.7637
	1.9038	1.9038	1.9038	1.9038	1.9216	1.9216	1.9216	1.9216	2.0431	2.0431	2.0431	2.0431	3.6768	3.6768	3.6768	3.6768	4.3314	4.3314	4.3314	4.3314
20	2.2904	2.2900	2.0164	2.0160	2.2840	2.2845	2.0183	2.0178	2.3286	2.3306	2.0436	2.0439	3.3546	3.3510	2.8833	2.8974	4.0680	4.1992	3.7802	4.0108
	2.2904	2.2900	2.0164	2.0160	2.2835	2.2837	2.0207	2.0200	2.3325	2.3344	2.0541	2.0539	3.8289	3.8376	3.4606	3.4829	4.4322	4.5570	4.4257	4.5827
16	2.7343	2.7330	2.6563	2.6492	2.7111	2.7092	2.6299	2.6249	2.7201	2.7153	2.6060	2.6101	3.6528	3.6186	3.4987	3.6080	4.2165	4.3939	4.1805	4.5921
	2.7343	2.7330	2.6563	2.6492	2.7101	2.7093	2.6337	2.6280	2.7162	2.7143	2.6179	2.6229	3.9935	3.9656	3.9457	4.1021	4.5369	4.6939	4.8155	5.2028
10	4.2619	4.2264	3.6668	3.6285	4.1982	4.1574	3.6059	3.5672	4.1033	4.0525	3.5292	3.4996	4.4513	4.3942	4.3143	4.4736	4.7680	5.0076	5.0298	5.5304
	4.2619	4.2264	3.6668	3.6285	4.2104	4.1685	3.6168	3.5769	4.0990	4.0611	3.5414	3.5081	4.5641	4.5351	4.5522	4.7526	4.9112	5.0607	5.3199	5.8070
4	7.3221	7.0559	7.1854	6.9266	7.1722	6.9298	6.8943	6.6757	6.8828	6.6347	6.3960	6.2551	6.4763	5.9754	6.1215	5.9377	6.4115	5.9093	5.9452	6.8017
	7.3221	7.0559	7.1854	6.9266	7.2244	6.9599	6.9491	6.7203	6.9913	6.6918	6.4606	6.2994	6.5064	5.9770	6.1426	5.9703	6.2969	5.8874	5.9588	6.9115
0	10.2750	9.9195	10.1070	9.3935	10.1250	9.6872	9.7004	9.1482	9.8153	9.1883	9.0112	8.7408	9.1246	7.5996	8.7273	8.3649	8.5931	6.8335	8.6620	7.9754
	10.2750	9.9195	10.1070	9.3935	10.1870	9.7641	9.8304	9.2068	9.9613	9.3294	9.1974	8.8168	9.3033	7.6983	8.7582	8.4108	8.7016	6.8716	8.7015	7.9640

Table 5.9 The average RMSE values for LZ time series prediction for trapezoidal MFs (shaded values are the results of SP-CB method).

	c=0				c=0.2				c=0.4				c=0.8				c=1.0			
	NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain	
	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS
NFtest	1.8722	1.8722	1.8722	1.8722	1.9014	1.9014	1.9014	1.9014	2.0430	2.0430	2.0430	2.0430	2.9051	2.9051	2.9051	2.9051	3.6759	3.6759	3.6759	3.6759
	1.8722	1.8722	1.8722	1.8722	1.8854	1.8854	1.8854	1.8854	1.9731	1.9731	1.9731	1.9731	3.1841	3.1841	3.1841	3.1841	3.7066	3.7066	3.7066	3.7066
20	2.3453	2.3442	1.9901	1.9887	2.3361	2.3353	2.0018	1.9996	2.3867	2.3879	2.0809	2.0777	3.3328	3.3745	2.9421	2.9966	4.0494	4.2312	3.8521	4.1303
	2.3453	2.3442	1.9901	1.9887	2.3368	2.3363	1.9936	1.9921	2.3574	2.3587	2.0290	2.0269	3.5321	3.5491	3.1343	3.1551	4.0926	4.2811	3.9341	4.2050
16	2.8237	2.8212	2.5962	2.5894	2.7951	2.7930	2.5418	2.5375	2.8051	2.8030	2.4869	2.4910	3.6726	3.7057	3.1981	3.3582	4.2920	4.4693	4.0887	4.6965
	2.8237	2.8212	2.5962	2.5894	2.8047	2.8031	2.5673	2.5634	2.7919	2.7909	2.5073	2.5129	3.8138	3.8098	3.4341	3.5864	4.2953	4.4787	4.1295	4.7351
10	4.4105	4.3712	3.6893	3.6487	4.3539	4.3062	3.6757	3.6535	4.2569	4.2177	3.6542	3.6148	4.5405	4.5615	4.3725	4.4819	4.8854	5.0474	4.9239	5.2322
	4.4105	4.3712	3.6893	3.6487	4.3713	4.3325	3.6655	3.6257	4.2830	4.2460	3.6518	3.6236	4.5862	4.5895	4.3735	4.4565	4.8778	5.0321	4.8997	5.1934
4	7.5223	7.2459	7.5623	7.3612	7.3929	7.1266	7.3392	7.1732	7.1625	6.8683	6.9936	6.9101	6.8087	6.2660	6.9578	6.8336	6.7201	6.1525	6.7943	7.4820
	7.5223	7.2459	7.5623	7.3612	7.4569	7.1759	7.4288	7.2551	7.2616	6.9616	7.1076	7.0104	6.8905	6.3078	6.8957	6.8479	6.8191	6.2788	6.8069	7.3902
0	10.4640	10.0690	9.8415	9.1247	10.3550	9.8780	9.4685	8.8814	10.0990	9.4597	8.9122	8.5988	9.5512	8.0315	8.7590	8.4203	9.3635	7.1663	8.6901	7.9620
	10.4640	10.0690	9.8415	9.1247	10.4150	9.9549	9.6426	9.0050	10.1930	9.6115	9.1544	8.7439	9.7151	8.2182	8.8689	8.4785	9.6822	7.5454	8.6935	8.1244

Table 5.10 The average RMSE values for LZ time series prediction for Gaussian MFs (shaded values are the results of SP-CB method).

	c=0				c=0.2				c=0.4				c=0.8				c=1.0			
	NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain		NFtrain		Ntrain	
	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS	SFLS	NSFLS
NFtest	1.9718	1.9718	1.9718	1.9718	1.9921	1.9921	1.9921	1.9921	2.1278	2.1278	2.1278	2.1278	3.4008	3.4008	3.4008	3.4008	4.3616	4.3616	4.3616	4.3616
	1.9718	1.9718	1.9718	1.9718	1.9883	1.9883	1.9883	1.9883	2.0888	2.0888	2.0888	2.0888	3.3257	3.3257	3.3257	3.3257	4.1932	4.1932	4.1932	4.1932
20	2.4037	2.4030	2.0263	2.0262	2.3954	2.3961	2.0282	2.0287	2.4775	2.4821	2.0997	2.1032	3.6882	3.6954	3.1481	3.1566	4.5659	4.5730	4.1996	4.2112
	2.4037	2.4030	2.0263	2.0262	2.4004	2.4005	2.0334	2.0338	2.4474	2.4497	2.0834	2.0851	3.5949	3.6014	3.0563	3.0636	4.3697	4.3774	3.9798	3.9908
16	2.8637	2.8596	2.5399	2.5329	2.8375	2.8358	2.5084	2.5033	2.8840	2.8861	2.5666	2.5722	3.9221	3.9330	4.1601	4.1834	4.6722	4.6921	5.2666	5.2934
	2.8637	2.8596	2.5399	2.5329	2.8496	2.8473	2.5281	2.5237	2.8626	2.8654	2.5652	2.5684	3.8252	3.8390	4.0887	4.1089	4.5061	4.5247	5.0767	5.0998
10	4.3816	4.3537	3.6064	3.5726	4.3308	4.2957	3.5643	3.5360	4.2671	4.2362	3.5570	3.5731	4.6267	4.6425	4.7117	4.7729	4.9422	4.9574	5.5488	5.6140
	4.3816	4.3537	3.6064	3.5726	4.3471	4.3233	3.5759	3.5508	4.2785	4.2656	3.5480	3.5476	4.6017	4.6181	4.5782	4.6308	4.9120	4.9234	5.3540	5.4133
4	7.2989	7.1463	6.9223	6.6560	7.1714	7.0209	6.6967	6.4440	6.8214	6.7382	6.2843	6.1456	6.0090	5.9382	5.9713	5.9585	5.6999	5.6033	6.1320	6.2235
	7.2989	7.1463	6.9223	6.6560	7.2219	7.0765	6.7600	6.5332	7.0058	6.8749	6.3997	6.2696	6.2890	6.1458	6.0051	5.9986	5.9668	5.8120	6.0558	6.1139
0	10.2730	9.8625	9.6172	9.0329	9.9039	9.6539	9.3456	8.8139	9.2875	9.1967	8.8264	8.5740	7.6377	7.4621	8.3701	8.3167	6.5412	6.3194	7.8453	7.7800
	10.2730	9.8625	9.6172	9.0329	10.1370	9.7466	9.4382	8.9399	9.7665	9.4048	9.0351	8.7372	8.4199	7.8862	8.5249	8.4175	7.6477	7.0138	8.3207	8.1771



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.8 Sample FLSs outputs for MG TSP by applying the SP-CB method with different settings. Tested with 20dB and FOU size $c = 0.2$, (a) trained with NF data and (b) trained with 20dB. Tested with 20dB and FOU size $c = 0.8$, (c) trained with NF data and (d) trained with 20dB. Tested with 10dB and FOU size $c = 0.2$, (e) trained with NF data and (f) trained with 10dB. Tested with 10dB and FOU size $c = 0.8$, (g) trained with NF data and (h) trained with 10dB.

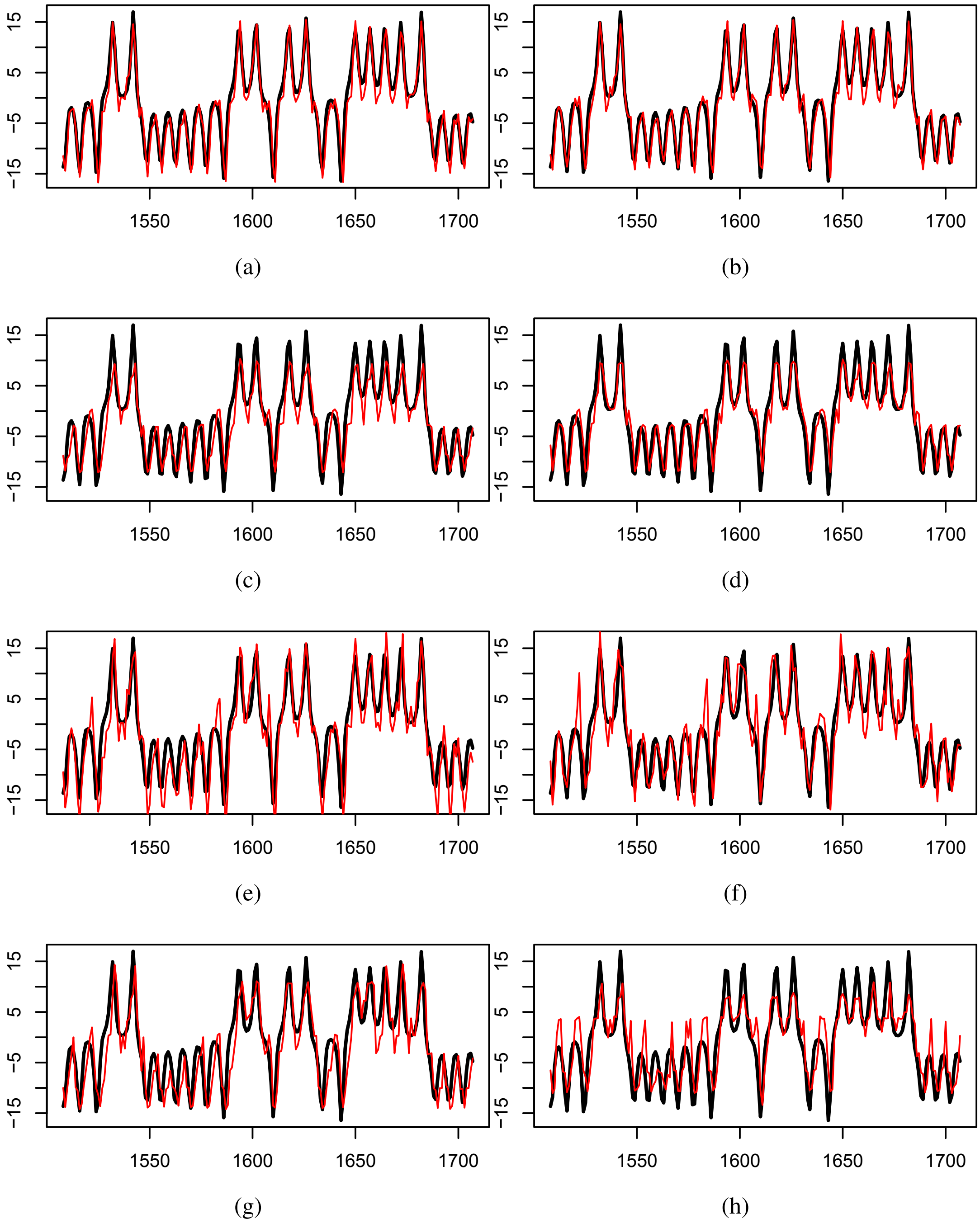


Figure 5.9 Sample FLSs outputs for LZ TSP by applying the SP-CB method with different settings. Tested with 20dB and FOU size $c = 0.2$, (a) trained with NF data and (b) trained with 20dB. Tested with 20dB and FOU size $c = 0.8$, (c) trained with NF data and (d) trained with 20dB. Tested with 10dB and FOU size $c = 0.2$, (e) trained with NF data and (f) trained with 10dB. Tested with 10dB and FOU size $c = 0.8$, (g) trained with NF data and (h) trained with 10dB.

5.4.1 Mackey-Glass Time Series Results

The average RMSE values for CB and SP-CB using MG time series are depicted in Tables 5.5 - 5.7. The results for the triangular, trapezoidal and Gaussian MFs are visualised in Figures 5.10, 5.11 and 5.12 respectively. Generally, there are no significant differences between them in the sense that NSFLSs provide superior outputs to SFLSs. Also, training the systems on noisy data (Ntrain) produces better performance especially with the SFLSs cases. As we increase the FOU size parameter c, the performance gets better. At $c = 0$ both methods have the same result as the systems reduced to the original T1 FLS.

By calculating the percentage of improvement of CB over SP-CB in the triangular MF case, is found to be 34% at $c = 0.8$ and 20% at $c = 1.0$. The percentage of improvement of CB over SP-CB in the case of the trapezoidal MF is found to be around 22% at $c = 0.8$ and less for the other c values. Whereas, The percentage of improvement of CB over SP-CB in the case of the Gaussian MF is smaller, reaching around 11.5% at $c = 1.0$ and significantly less for the other c values.

A general comparison of CB and SP-CB performance in different setting (e.g., training on NFtrain and Ntrain data, SFLSs and NSFLSs)) is presented next. By comparing the results of each methods, we have found that the FLSs trained with noisy data (Ntrain) performed better than those trained with noise free (NFtrain) in all MFs cases, as expected (see for example, Figures 5.10(a) and 5.10(e)). In the case of SFLSs and at low SNR (higher noise levels), FLSs trained with noisy data (Ntrain) perform better by around 41 % than FLSs trained with NF data (NFtrain). From Figures 5.10-5.12, it is clear that the CB method is performing better than the

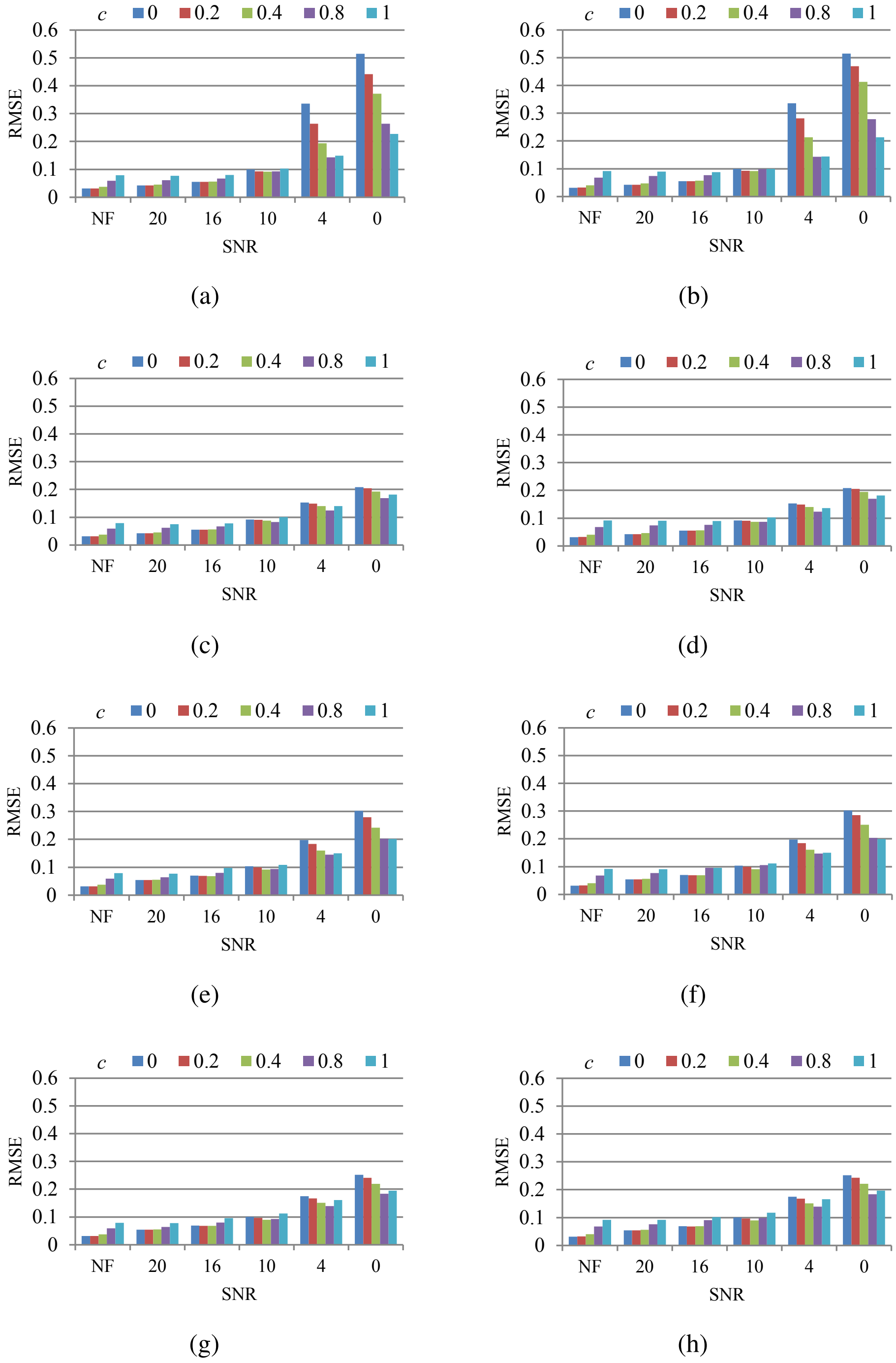


Figure 5.10 The average RMSE of the TSP for the MG TSP and triangular MFs. (a) CB method and (b) SP-CB method for SFLSs training on NF data (NFtrain); (c) CB method and (d) SP-CB method for NSFLSs trained on NF data (NFtrain); (e) CB method and (f) SP-CB method for SFLSs trained on noisy data (Ntrain); (g) CB method and (h) SP-CB method for NSFLSs trained on noisy data (Ntrain).

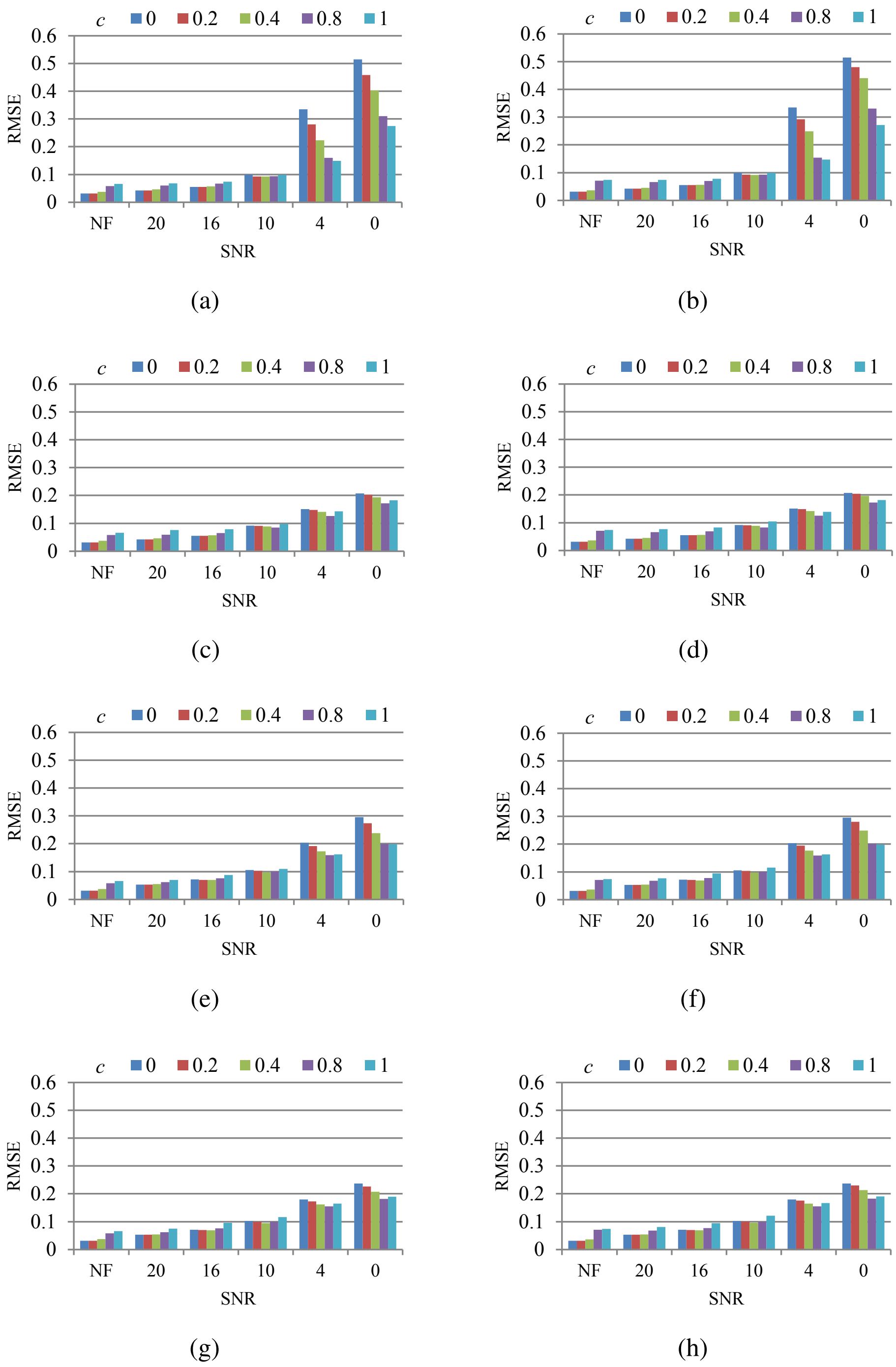


Figure 5.11 The average RMSE of the TSP for the MG TSP and trapezoidal MFs. (a) CB method and (b) SP-CB method for SFLSs training on NF data (NFtrain); (c) CB method and (d) SP-CB method for NSFLSs trained on NF data (NFtrain); (e) CB method and (f) SP-CB method for SFLSs trained on noisy data (Ntrain); (g) CB method and (h) SP-CB method for NSFLSs trained on noisy data (Ntrain).

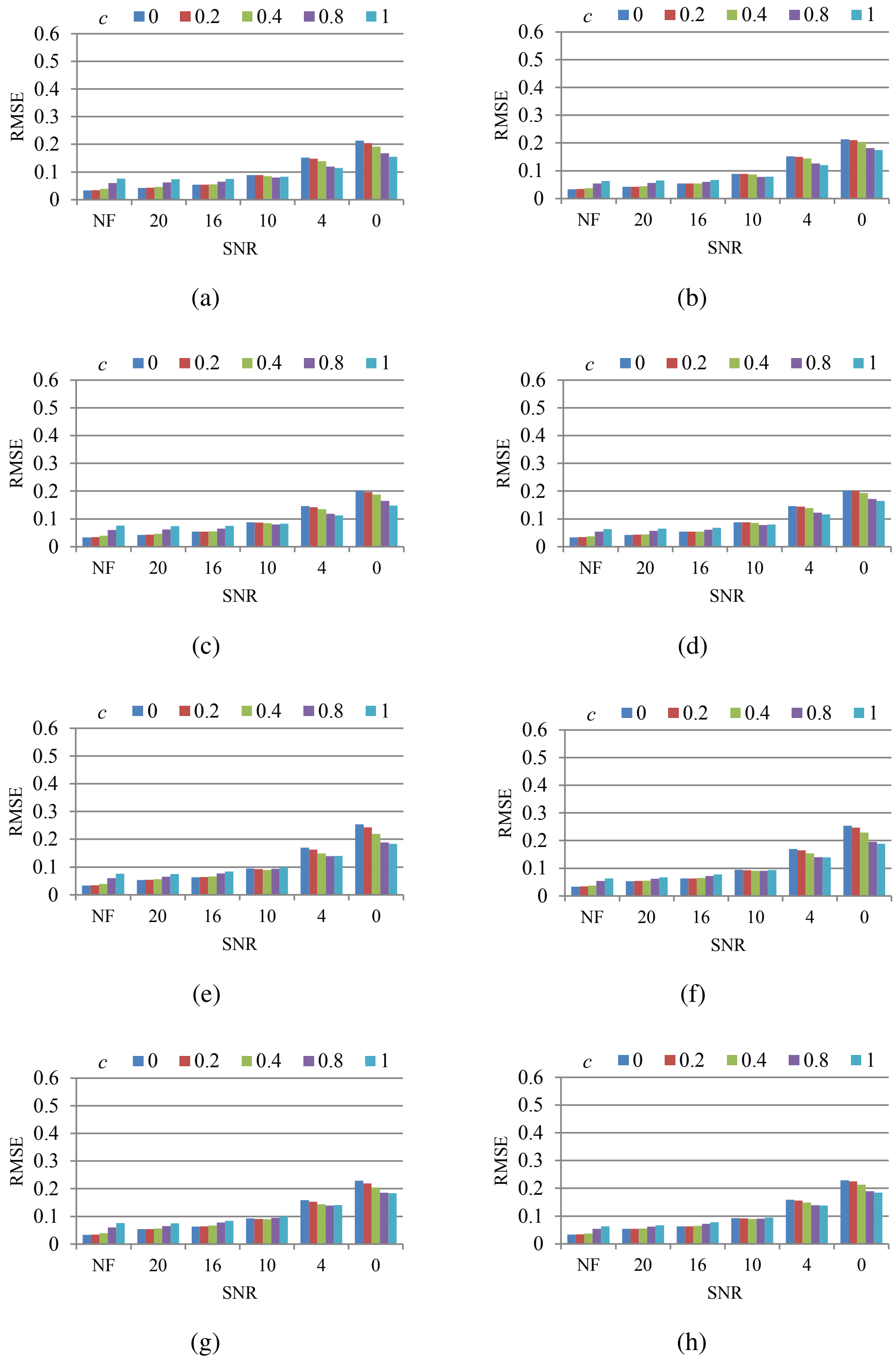


Figure 5.12 The average RMSE of the TSP for the MG TSP and Gaussian MFs. (a) CB method and (b) SP-CB method for SFLSs training on NF data (NFtrain); (c) CB method and (d) SP-CB method for NSFLSs trained on NF data (NFtrain); (e) CB method and (f) SP-CB method for SFLSs trained on noisy data (Ntrain); (g) CB method and (h) SP-CB method for NSFLSs trained on noisy data (Ntrain).

SP-CB method in most cases. Overall, as expected, NSFLSs perform much better than SFLSs, especially at lower SNR values (higher noise levels).

By comparing the result of triangular, trapezoidal and Gaussian MFs performance in the case of both the CB and SP-CB shown in Tables 5.5 - 5.7 together with Figures 5.10- 5.12, we have found that FLSs with trapezoidal MFs perform better than those with triangular MFs in all cases except for SFLSs trained on NF data and Gaussian MFs results show superior performance. Also, NSFLSs perform much better than SFLSs in both training cases (noise free (NFtrain) and noisy (Ntrain) training data).

5.4.2 Lorenz Time Series Results

The average RMSE values for CB and SP-CB using LZ time series are depicted in Tables 5.8-5.10. For a better illustration of the results in Tables (5.8-5.10), we show a visual representation of the these results in Figures 5.13-5.15. These figures show the RMSE values of SFLSs and NSFLSs using different noise levels (different SNR values) of testing and training data corrupted with different levels of noise for the LZ time series. These figures, together with Tables (5.8-5.10), show the performance of each FLS designed with a particular FOU size (parameter c) and trained with particular noise level, as each is tested with different noise level. As can be seen from the tables and their corresponding figures, the results for the LZ TSP are consistent with the results of the MG TSP and both lead to the same observations and conclusions detailed further below.

Generally, NSFLSs provide superior outputs to SFLSs and there are no significant differences between the performance of both approaches (CB and SP-CB). Also, training the systems on noisy data (Ntrain) produces better performance especially

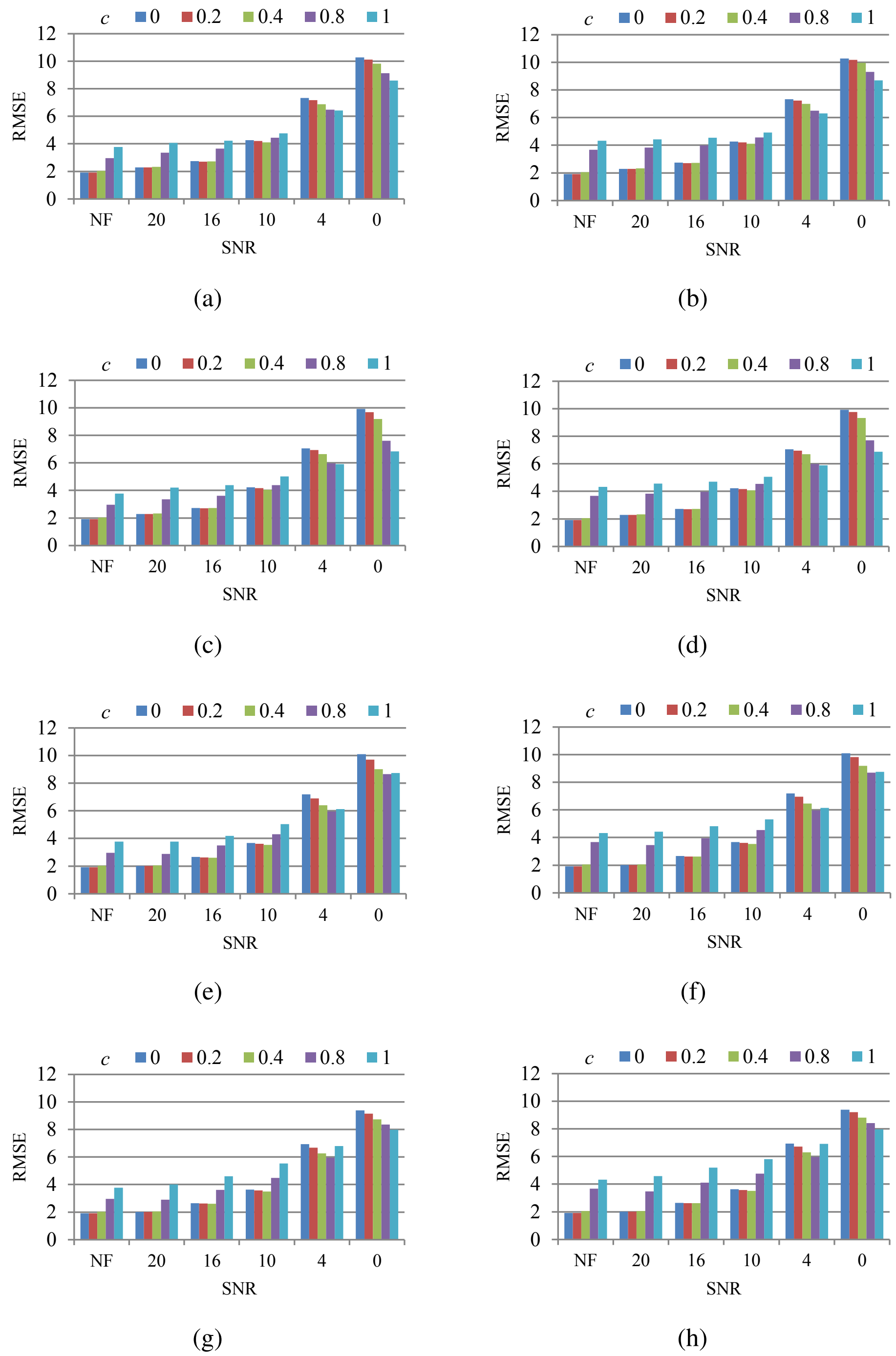


Figure 5.13 The average RMSE of the TSP for the LZ TSP and triangular MFs. (a) CB method and (b) SP-CB method for SFLSs training on NF data (NFtrain); (c) CB method and (d) SP-CB method for NSFLSs trained on NF data (NFtrain); (e) CB method and (f) SP-CB method for SFLSs trained on noisy data (Ntrain); (g) CB method and (h) SP-CB method for NSFLSs trained on noisy data (Ntrain).

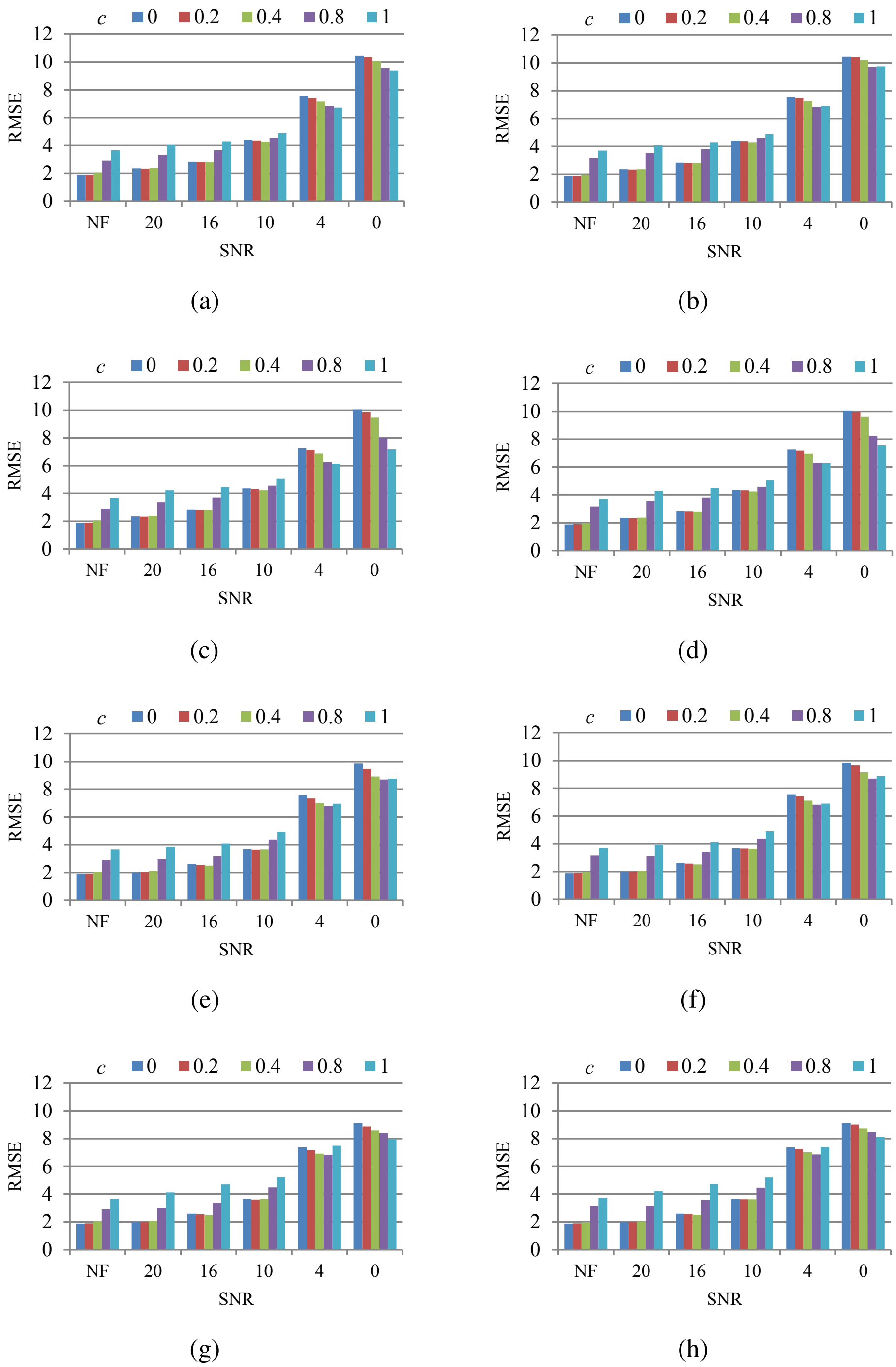


Figure 5.14 The average RMSE of the TSP for the LZ TSP and trapezoidal MFs. (a) CB method and (b) SP-CB method for SFLSs training on NF data (NFtrain); (c) CB method and (d) SP-CB method for NSFLSs trained on NF data (NFtrain); (e) CB method and (f) SP-CB method for SFLSs trained on noisy data (Ntrain); (g) CB method and (h) SP-CB method for NSFLSs trained on noisy data (Ntrain).

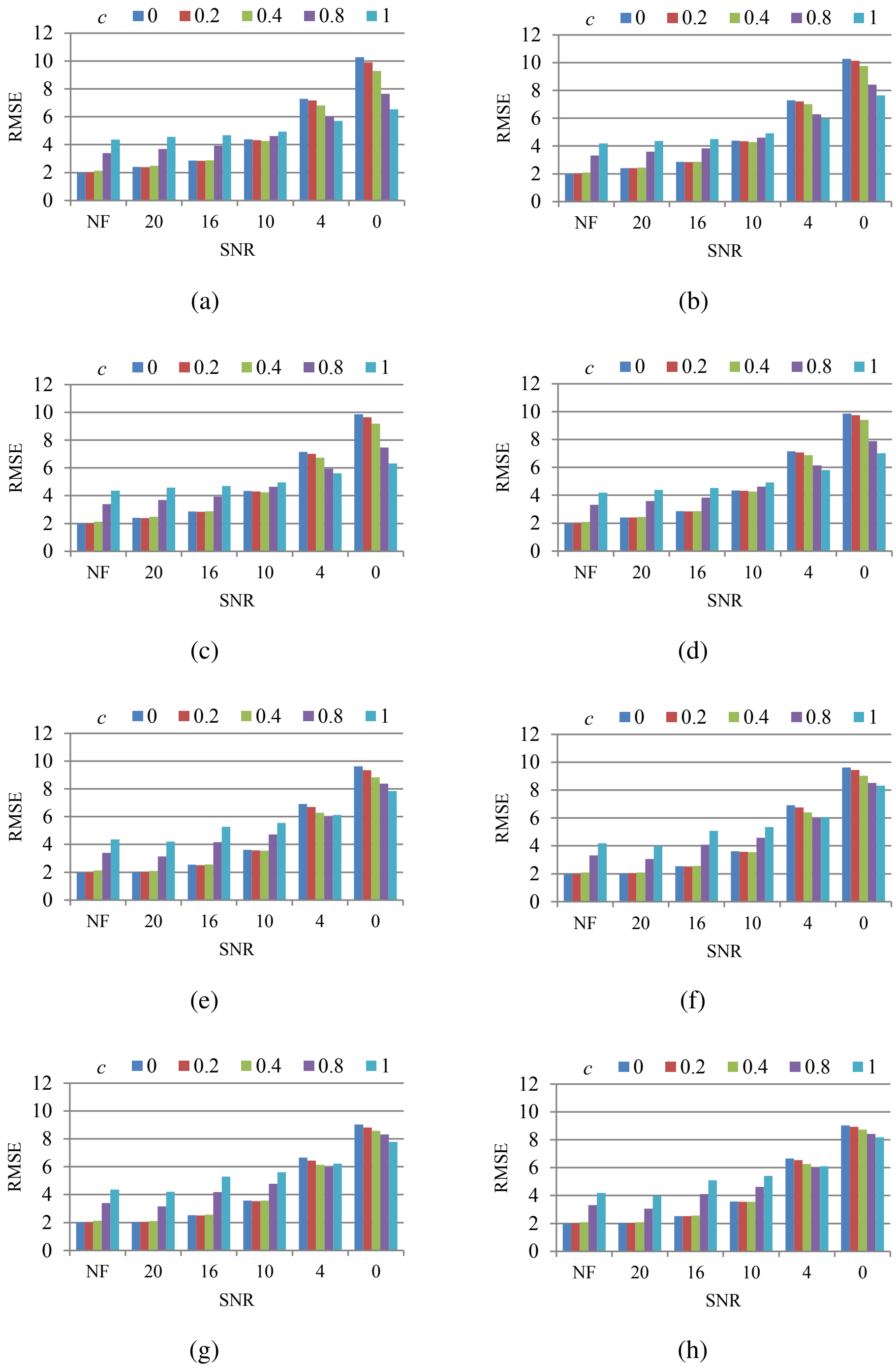


Figure 5.15 The average RMSE of the TSP for the LZ TSP and Gaussian MFs. (a) CB method and (b) SP-CB method for SFLSs training on NF data (NFtrain); (c) CB method and (d) SP-CB method for NSFLSs trained on NF data (NFtrain); (e) CB method and (f) SP-CB method for SFLSs trained on noisy data (Ntrain); (g) CB method and (h) SP-CB method for NSFLSs trained on noisy data (Ntrain).

with the SFLSs cases. As we increase the FOU size parameter c, the performance gets better. At $c = 0$ both methods (CB and SP-CB) have the same result as the systems reduced to the original T1 FLS.

By calculating the percentage of improvement for the LZ time series of CB over SP-CB in the triangular MF case, is found to be 21% at $c = 0.8$ and 14% at $c = 1.0$. The percentage of improvement of CB over SP-CB in the case of the trapezoidal MF is found to be around 10% at $c = 0.8$ and less for the other c values. However, the percentage of improvement of CB over SP-CB in the case of the Gaussian MF is smaller, reaching around 5.5% at $c = 1.0$ and significantly less for the other c values.

A general comparison of CB and SP-CB performance in different settings (e.g., training on NFtrain and Ntrain data, SFLSs and NSFLSs)) is presented next. By comparing the results of the methods, we have found that the FLSs trained with noisy data (Ntrain) performed better than those trained with noise free (NFtrain) in all MF cases, as expected (see for example, Figures 5.14(a) and 5.14(e)). In the case of SFLSs and at low SNR (higher noise levels), FLSs trained with noisy data (Ntrain) perform better by around 18% than FLSs trained with NF data (NFtrain). From Figures 5.13-5.15, it is clear that the CB method is performing better than the SP-CB method in most cases. Overall, as expected, NSFLSs perform much better than SFLSs, especially at lower SNR values (higher noise levels).

By comparing the performance results of LZ TSP using triangular, trapezoidal and Gaussian MFs in the case of both the CB and SP-CB shown in Tables 5.5 - 5.7 together with Figures 5.10-5.12, we have found that FLSs with the trapezoidal MFs perform better than those with the triangular MFs in all cases and Gaussian MFs results show superior performance. Also, NSFLSs perform much better than SFLSs in both training cases (noise free (NFtrain) and noisy (Ntrain) training data).

Finally, from the results of both time series, we observe the following:

- All FLSs show performance improvement when SNR increases as is intuitive.
- A direct relationship between the FOU size of the FSs and the noise level is observed showing that as the noise level increases, the FOU that gives the minimum RMSE value increases as well, resulting in performance improvement.
- Training the systems with noisy data (Ntrain) improves the performance of FLSs.
- NSFLSs outperform their counterparts (SFLSs).
- The Wang-Mendel approach generates the same number of rules for triangular, trapezoidal and Gaussian MFs at each noise level for each different time series.
- The CB method shows better performance than the SP-CB method, especially with triangular MFs.
- Gaussian MFs show better performance overall compared to the other two MFs (triangular and trapezoidal).

5.5 Summary

In this chapter, we propose the SP-CB approach to transitioning from T1 to IT2 FSs for different levels of uncertainty (noise), while preserving the original T1 MF shape in the IT2 MFs thus satisfying requirements 1, 2 and 3 in Section 3.2. The objective of this work is not to achieve optimal performance in applications such as in time series prediction, but to study and present an IT2 FS creation method. The SP-CB method systematically captures a specified amount of uncertainty (i.e., the uncertainty in memberships over the support of the LMF is constant) and preserves the original shape of the MF (the LMF and UMF keep their original T1 MF shape) for comparison between T1 and IT2 FLSs. The proposed approach (SP-CB) enables both the adaptation of the IT2 FS for known levels of uncertainty (i.e. by increasing FOU size with increasing uncertainty) and the systematic comparison of the original T1 FLS(s) to the resulting new IT2 FLS(s). This method is compared with the IT2 FS creation method CB, which follows a similar creation approach as SP-CB but does not maintain the T1 MF shape.

In order to assess the viability and explore the behaviour of the SP-CB method, we conducted a detailed performance comparison and evaluation in the context of time series analysis using both MG and LZ time series. Both methods were tested under different conditions (Noise free and noisy training data, singleton and non-singleton fuzzification) as well as using triangular, trapezoidal and Gaussian MFs. Generally, the results indicate that FLSs based on the CB method outperform those based on the SP-CB method in both time series. However, both methods provide expected performance increases for increasing FOU sizes as uncertainty/noise levels increase. Based on this, it seems that in applications where the systematic transition and comparison from an original T1 FLS is paramount, the SP-CB method is preferable, as it maintains the MF shapes. In applications where this level of comparability is not vital however, the CB method provides superior levels of performance, while also maintaining a systematic and parametrised increase in FOU size in the face of increasing uncertainty.

Based on the results achieved in this investigation, the adaptation to the systematic design and evaluation process and the conclusions drawn from applying and comparing the results of CB and SP-CB methods in the given application, the first and the second objectives stated in the introduction are met in this chapter.

The following chapter will present the third approach investigated in this thesis. This approach is built on the the first presented method (CB method) and is used to further refine the FOU size for an existing system by introducing an optimisation method for tuning IT2 FLSs parameters and selecting the optimal FOU through the simulated annealing algorithm.

Chapter 6

An Exploration into the Optimised-Controlled Blurring for the Tuning of Interval Type-2 Fuzzy Sets

6.1 Introduction

It is known that the uncertainty in an application varies if the environment or conditions change. Assuming that we have established an appropriate FOU in the existing system and the environment or conditions change, in this case, the FOU refinement is crucial to enable better adaptation in the new condition.

This chapter explores the third approach, namely, the optimised controlled blurring (O-CB) method initially introduced in Chapter 3 (Section 3.2.4), used to refine the FSs for existing systems and to select the optimal FOU size in a given application faced with varying levels of uncertainty in order to enable better adaptation in real world applications by exploring an optimisation method (e.g., simulated annealing). This approach satisfying requirements 1, 2 and 4 presented in Chapter 3 (Section 3.2).

The main feature of T2 FSs is their ability to represent uncertainties within a system. These uncertainties are captured in the FOU of a T2 MF which can be described by the upper and the lower membership function. One of the challenges in modelling a T2 FLS is the problem of defining the MF parameters and their FOUs, given noisy data or imperfect measurements. This challenge is increased by the complexity which arises from the increase in the number of parameters of IT2 MFs to be tuned.

Designing and tuning IT2 FLSs parameters and selecting the optimal FOU are common techniques used to improve the performance of systems. However, the process of manually designing and tuning IT2 FLSs is complicated because these systems have more parameters to adjust than their T1 FLSs counterparts, especially with the increase of the MF parameters [140]. Thus, automatic optimisation methods such as those based on neural networks (NN) [144],[145], genetic algorithms (GA) [140],[146–149] or simulated annealing (SA) [150–156] have been used in the design of IT2 FLSs in many areas of application in an effort to increase the performance of systems by finding 'suitable' FOUs.

To achieve the fourth objective stated in Chapter 1 (to refine the FSs for existing systems and select the optimal FOU size in a given application faced with varying levels of uncertainty in order to enable a better adaptation in real world applications), in this chapter, a method for tuning IT2 FLS is adopted again here, and then the design parameters are tuned through the simulated annealing (SA) optimisation algorithm. The CB method has few parameters to be tuned than the conventional approach, as only a single extra parameter is used to define the IT2 MFs. Since we are investigating a specific problem (i.e., refining the FOU size of IT2 FS), we demonstrate the approach on one benchmark time series prediction problem; the MG TSP problem, using training data sets corrupted with different levels of noise and two different MFs: triangular and Gaussian. By doing so, we demonstrate that this approach is an appropriate FOU selection mechanism that produces IT2 FLSs with good performance.

In this work, SA optimisation (see Chapter 2 Section(2.9.2) for more details) is used to tune the IT2 FLSs designed with the controlled blurring (CB) method (detailed in Chapter 4). Although any global optimisation method such as GA can be used, we have chosen SA as it has been successfully used in fuzzy system applications and more specifically in the forecasting of MG time series [157–159, 173], as well as often requiring less CPU time than GA [174]. The proposed approach is applied to the forecasting of MG time series and its results are evaluated.

In the remaining sections of this chapter, we describe the detailed concepts of the optimised controlled blurring (O-CB) (in Section 6.2). A series of experiments that illustrate the O-CB approach and follow the IT2 FS optimising and evaluation process highlighted in Chapter 3 (Section 3.3) will be detailed in section 6.3 and the results will be presented. The results will be then analysed and discussed in section 6.4, followed by the chapter summary in section 6.5.

6.2 The Optimised Controlled Blurring Method Details

In this section, we show a proposed approach on the appropriate adjustment of the FOUs at run time, i.e., when uncertainty levels vary. Specifically, we explore the application of simulated annealing optimisation methods to refine the FOU sizes in IT2 FSs and hence, satisfying requirements 1, 2 and 4 presented in Chapter 3 (Section 3.2).

To simplify the complexity which arises from the increase in the number of parameters to be tuned in IT2 MFs, the FOU creation method CB is adopted here. The CB method is designed to create IT2 FSs with a uniform FOU over the support of the lower MF (LMF) 'core' of the fuzzy set by incorporating a fixed FOU size parameter $c \in [0,1]$ used to create an FOU of a given size around a T1 MF. This method has fewer parameters than usual IT2 FS. It is dependent partially on a T1 FS (i.e., usual T1 MF parameters), introducing only one extra parameter (FOU size parameter) as discussed in Chapter 4. An example of an FOU creation approach using this CB method is depicted in Figures 6.1 and 6.2 for the triangular and Gaussian MFs cases respectively, designed with the FOU sizes parameter c=0.4 and c=0.80 using equation (4.1) and (4.2). Starting from the initial T1 FS shown in Figures 6.1(a) and 6.2(a), two different IT2 FSs are created by adding the chosen FOU size parameters $c = 0.4$ in Figures 6.1(b) and 6.2(b) and $c = 0.8$ in Figures 6.1(c) and 6.2(c). The UMFs and LMFs are obtained using equations (4.1) and (4.2). For more details of this method we refer the reader to Chapter 4.

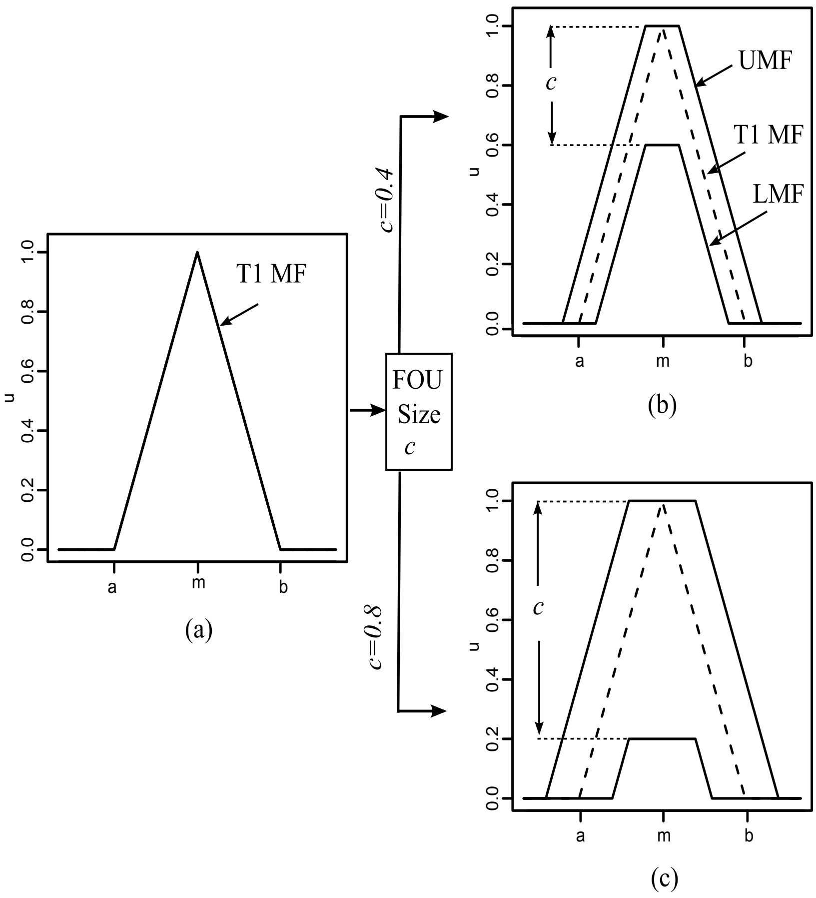


Figure 6.1 Illustration of the FOU design of IT2 triangular MFs using CB method. (a) Initial T1 FS, (b) IT2 obtained using FOU size parameter $c = 0.4$ and (c) IT2 FS obtained using FOU size parameter $c = 0.8$.

Therefore, an implementation of this FOU creation method (i.e., the CB method) by utilising an automatic optimisation method will increase the performance of the FLSs by finding ‘suitable’ FOUs and more adaptation in real world applications.

The proposed approach will use the SA algorithm (detailed in Chapter 2 Section 2.9.2) by first defining its parameters and functions including the initial solution obtained from the T1 system and the objective function. The optimisation method

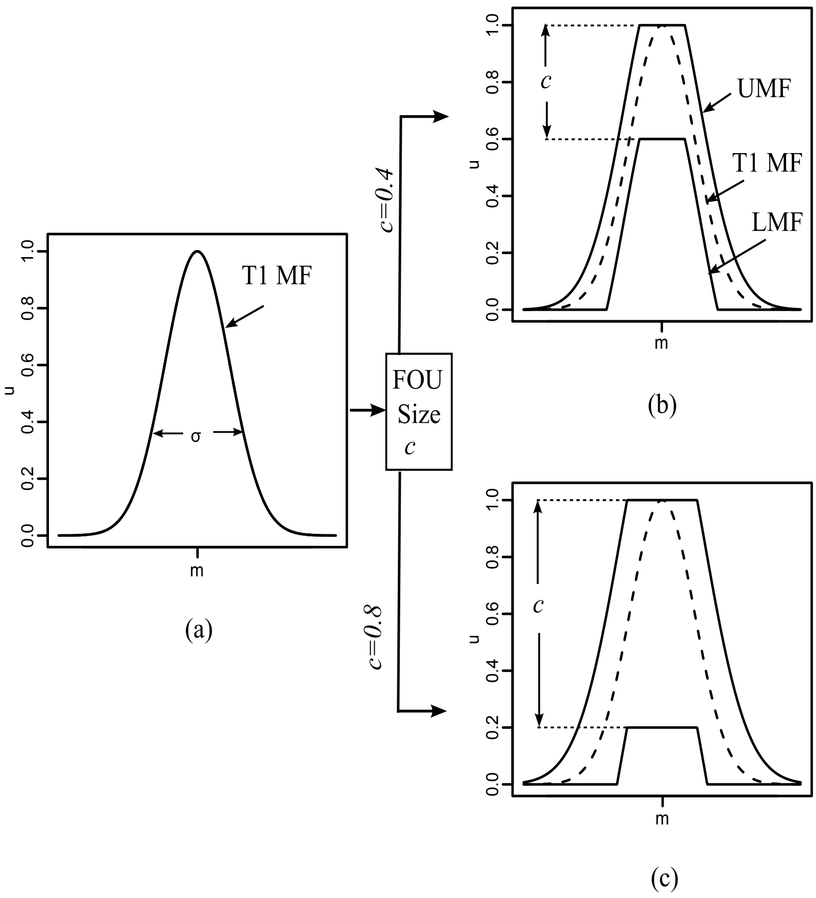


Figure 6.2 Illustration of the FOU design of IT2 Gaussian MFs using CB method. (a) Initial T1 FS, (b) IT2 obtained using FOU size parameter $c = 0.4$ and (c) IT2 FS obtained using FOU size parameter $c = 0.8$.

iterates until the stopping criteria is reached and then producing the final FLS state that will be used as the optimal system parameters including the FOU size.

6.2.1 Initial Solution

The initial solution is based on the training data. The initial T1 FSs in (4.1) and (4.2) are created using the training data maximum and minimum values to divide each input space into the chosen number of fuzzy sets, providing sufficient overlapping between them. The FOU size parameter c will be set initially to 0.

The T1 FSs are extended to IT2 FSs using the existing T1 MFs parameters (the initial solution) as a basis and the FOU size parameter $c \in [0,1]$. The UMFs and LMFs are created using (4.1) and (4.2) respectively.

6.2.2 Simulated Annealing Parameters

After defining the initial solution, S_i, the SA parameters are defined to start using the algorithm for the learning and tuning process. The choice of the simulated annealing parameters is important for its success. For example, the choice of small initial temperatures or cooling rate could result in local minimas solution, while the choice of large ones could cause a very long search in the solution space and more running times that make the algorithm very slow. As detailed in Chapter 2 (Section 2.9.2), the SA algorithm needs the following parameters to be specified.

1. Initial temperature T_i. Initial temperature T_i can be given by the user or can be calculated using the method described in Chapter 2 (Section 2.9.2) with implementation to (2.35) and (2.36).

2. Fitness function F. The fitness function (also called the objective function or the cost function) is a function used to measure the quality of the solution. To define the fitness function for the SA, Root Mean Square Error (RMSE) is used to evaluate the output result from the tuned FLS at the current temperature T. Generally, any objective function (fitness function) that can represent how close the FLSs' output to the actual output would be used. One of the most commonly used objective function in this case is an error function. This function is used to

```
SimAnn(S_i,T_i,α,MaxStg,MaxAnn)
     % S_i   initial Solution
     % T_i   initial Temperature
     % α   Cooling rate
     % MaxStg  Maximum temperature Stage length
     % MaxAnn  Maximum length for the Annealing process
    Begin
      T = T_i    %  Current Temperature
      S_curr = S_i    %  Current Solution
      S_bst = S_curr    % Best Solution seen so far
       Repeat
        Repeat
         CurrFit = F(S_curr)    % Current Solution Fitness using the function F
         BstFit = F(S_bst)    % Best Solution Fitness
         S_new = N(S_curr)    % generate New Solution using the function N  S_new near to S_curr
         NewFit = F(S_new)    % New Solution Fitness
         IF  NewFit < BstFit   THEN  S_bst = S_new
         IF  NewFit < CurrFit THEN  S_curr = S_new
          ELSE
          IF  Metropolis(NewFit,CurrFit,T) is TRUE
           THEN  S_curr = S_new
           ELSE  S_curr = S_curr
        Until MaxStg
         T = α * T
       Until MaxAnn
    Return S_bst
End
```

Figure 6.3 Simulated annealing pseudo code.

find the difference between the output of FLSs and the desired (actual) output by applying a statistical metric such as the RMSE. Then, the smallest fitness of the given solutions that are evaluated by the RMSE function will be taken as the best solution.

3. Neighbour solution N. For selecting a new solution, S_{new}, near the current one, S_{curr}, in the specified search space, an interval defining the minimum and the maximum changes to each parameter (except FOU size parameter c) is set as $[-\delta, \delta]$ where $\delta = (\max(\text{training data}) - \min(\text{training data}))/200$. The FOU size parameter c will be selected randomly from the interval [0,1].

4. Cooling rate α. Cooling rate, α, is a constant between 0 and 1 chosen to allow the temperature decrease according to a geometric decrement, as $T = \alpha T$.

```
Metropolis(NewFit,CurrFit,T)
      % CurrFit   Current Solution Fitness
      % NewFit   New Solution Fitness
      % T   Current Temperature
   Begin
    ΔF = NewFit - CurrFit   % change in Fitness ΔF = F(S_new) − F(S_curr)
    P = e^(−ΔF/T)   % acceptance probability
    Rand= (0,1)   % A random number in the range 0 to 1
    IF  Rand < P  THEN
     Return THEN
    ELSE
     Return FALSE
End
```

Figure 6.4 Metropolis acceptance procedure.

5. Maximum temperature stage length (MaxStg). It is a constant number of iterations at each temperature. This number can be related to the number of parameters to be tuned, for example, *number of parameters* x 2.

6. Maximum length for the annealing process (MaxAnn). Possible stopping criteria include when the temperature reaches zero or when a pre-specified number of iterations is reached.

The pseudo code of simulated annealing algorithm and the Metropolis acceptance procedure used to tune FLSs is shown in Figures 6.3 and 6.4 respectively.

After defining the SA parameters, the optimisation method iterates until the stopping criteria is reached producing the final FLS state that will be used as the optimal system parameters including the FOU size. We will refer to this approach as the optimised controlled blurring (O-CB) method throughout the thesis.

In the following section we conduct a set of experiments using the FOU refinement approach (O-CB) described in the context of the well-known MG time series.

6.3 Experiments and Results

In this section, we use the time series prediction as platforms to explore the behaviour of the FLSs tuned with our approach (O-CB), in respect to different levels of uncertainty/noise. We conduct a series of experiments using the MG time series, a chaotic time series; their details were given in Chapter 4 Section (4.3.1).

Following Chapter 3 (Section 3.3) and by applying the O-CB method using SA algorithm, each of the process steps is detailed below and applied to the MG time series. For MG TSP, we show two cases based on different type of FSs: triangular and Gaussian MFs. The results are then presented. For each case of the triangular and Gaussian MFs, we perform eight different experiments to learn and optimise eight different IT2 FLSs at four different levels of noise (SNRs)

First, we generate a data set (both training and testing data) from the MG time series. Then, we design evenly distributed T1 FSs for each MFs cases (triangular and Gaussian) and create the rule bases. Next, we start the tuning of a series of IT2 FLSs by using an SA algorithm based on the CB method to produce optimised IT2 FLSs with optimal FOU sizes. The actual number of FSs and the rules are maintained from the T1 system. In parallel, different levels of noise as a source of uncertainty are employed to generate a series of testing data sets. Finally, the performance of each of the T1/IT2 FLSs is evaluated for each of the testing data sets.

The complete process is illustrated by the flowchart in Figure 6.5 and can be summarised in the four subsections (6.3.1 - 6.3.4) below.

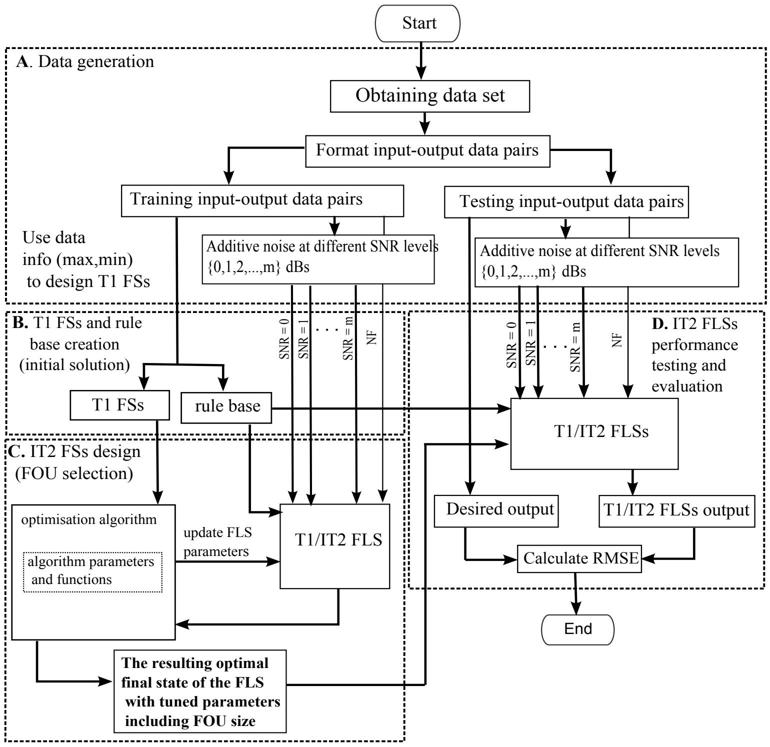


Figure 6.5 A flowchart of the process of using O-CB method to tune and evaluate IT2 FLSs at different noise levels.

6.3.1 Data Generation

Following Chapter 3 (Section 3.3.1), NF data are generated using (**??**) for the MG time series with the parameters and the numerical solutions of the differential equation presented in Chapter 4, Section (4.3.1. To obtain training and testing input-output data pairs, we extract 700 input-output data pairs. In our case, we consider four-input, one-output FLSs (single stage prediction). The training dataset is the first 500 data points used for learning and optimisation of the FLSs parameters using $x(1001)$ to $x(1504)$. The first input-output pair is of the form:

$$[input : x(1001), x(1002), x(1003), x(1004), \;\; output : x(1005)]$$

and the second pair is of the form:

$[input : x(1002), x(1003), x(1004), x(1005),\ \ output : x(1006)]$

and so on.

The next 200 points are used for testing the FLSs. In this work we consider two different testing data sets as follows:

1. Testing data set 1. This set is formed using $x(1505)$ to $x(1708)$ with the increment of the sequence equal to one, i.e. the first input-output pair is of the form:

   $[input : x(1505), x(1506), x(1507), x(1508),\ \ output : x(1509)]$

   and the second pair is:

   $[input : x(1506), x(1507), x(1508), x(1509),\ \ output : x(1510)]$

   and so on.

2. Testing data set 2. This set is formed using $x(1505)$ to $x(2504)$ with the increment equal to five, i.e. of the form:

   $[input : x(1505), x(1506), x(1507), x(1508),\ \ output : x(1509)]$

   and the second pair is:

   $[input : x(1510), x(1511), x(1512), x(1513),\ \ output : x(1514)]$

   and so on.

The objective of using two different sets is to reduce the chance of the cumulative error problem that occurs due to the prediction of successive points. The second testing data set offers a testing data with input-output pairs that are apart from each other and could lead to less prediction error. In this experiment, we consider different versions of training and testing data corrupted with zero-mean uniform noise for different SNRs.

We use 4 noise levels in training and the same number in testing. The chosen noise levels are (from lowest to highest level): 20, 10 and 0 dBs. The original noise free (NF) data are also used for training and testing the FLSs.

6.3.2 Type-1 Fuzzy Sets Design and Rule Base Creation

As mentioned above, the initial solution is defined based on the training data. For each case based on either triangular or Gaussian MFs, each of their four inputs and the output is designed with two MFs. In the case of the FLS that employed triangular MF, the initial T1 FS is characterised by a triangular MF defined by the mean m, left end point a and right end point b. Whereas, in the case of the FLS that employed Gaussian MF, the initial T1 FS is characterised by a Gaussian MF defined by the mean m and the standard deviation σ. The initial triangular and Gaussian T1 MFs used for the inputs and outputs are shown in Figure 6.6.

The means of each input and output are initialised by calculating the mean ($mean_{td}$) and the standard deviation (σ_{td}) of the training data set and setting each input and output MF parameters (for both triangular and Gaussian) as follows:

- For triangular MFs:

  $m_1 = mean_{td} - \sigma_{td}$; $m_2 = mean_{td} + \sigma_{td}$,

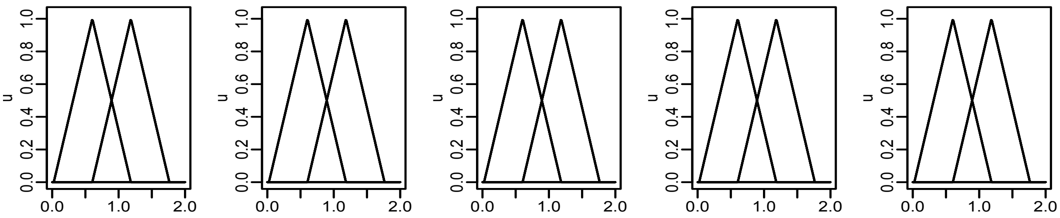
  $a_1 = m_1 - 2 * \sigma_{td}$; $a_2 = m_2 - 2 * \sigma_{td}$,

  $b_1 = m_1 + 2 * \sigma_{td}$; $b_2 = m_2 + 2 * \sigma_{td}$,

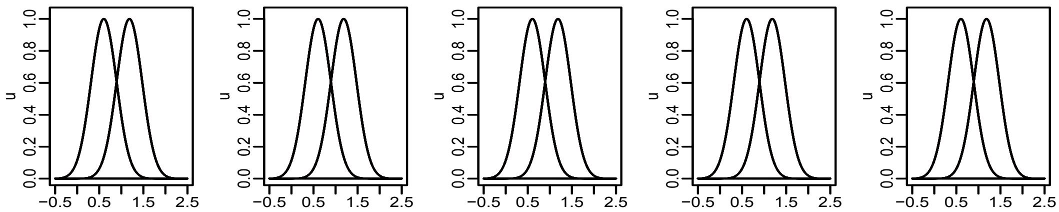
  where m_1, a_1, and b_1 are the parameters of the first FS and m_2, a_2, and b_2 are the parameters of the second FS. This enables sufficient overlap between the FSs.

- For Gaussian MFs:

  $m_1 = mean_{td} - \sigma_{td}$; $m_2 = mean_{td} + \sigma_{td}$,



(a)



(b)

Figure 6.6 The initial T1 MFs initialised for 4-inputs and 1-output.(a) Triangular MFs (b) Gaussian MFs .

σ_1=σ_2=σ_{td},

where m_1 and σ_1 are the parameters of the first FS and m_2 and σ_2 are the parameters of the second FS. This enables sufficient overlap between the FSs.

The tuning approach aims to optimise the parameters of the antecedents of the rules. The consequent part remains the same as the initialised T1 MF as we assume, there will be no uncertainty in the system output. By using 4-inputs and two FSs with each input, a total number of 16 rules (for both triangular and Gaussian cases) is found by using all possible combinations.

The rule base that is used during the learning and optimisation processes of the IT2 FLS is the T1 FLS rule base featuring IT2 MFs in the antecedents part, keeping the consequents as T1 MFs.

6.3.3 Interval Type-2 Fuzzy Sets Tuning (FOU Selection)

During the process of tuning the IT2 FLSs using SA, the T1 FSs are extended to IT2 FSs using the existing T1 MFs parameters (the initial solution) as a basis and the FOU size parameter $c \in [0, 1]$ (initially c will be set to 0). The UMFs and LMFs are created using (4.1) and (4.2) respectively using the FOU creation method (CB) detailed in Chapter 4. The actual number of FSs and the rules are maintained from the T1 system.

The simulated annealing (SA) algorithm described in Chapter 2 Section 2.9.2 is used here to optimise the IT2 FLSs based on the FOU creation method (CB) and to find the optimal FOU sizes. Following Section 6.2.2, the SA parameters are defined as follows:

1. The temperature T_i is initialised using (2.35) and (2.36).
2. The fitness function F is chosen to be the RMSE as shown in (3.2).
3. For the purpose of the neighbour solution, N, the interval that defines the minimum and the maximum changes to each parameter (except FOU size parameter c) is set as $[-\delta, \delta]$ where $\delta = 0.005$. The FOU size parameter, c, is selected randomly from the interval [0,1].
4. The cooling rate, α, is set at 0.90.
5. The maximum temperature stage length (MaxStg) is *number of parameters**2. Fore triangular MFs case, MaxStg = 50 stages and for Gaussian MFs case, MaxStg = 34 stages.
6. The maximum length for the annealing process (MaxAnn) is chosen to be 100 iterations.

In this work, we consider two cases of the FOU size selection. The first case keeps the FOU size the same for both FSs and the second case enables the SA algorithm to select a different size for each FS. The FLSs that used either triangular or Gaussian MFs consist of four inputs, each one employs two MFs and one output with two MFs. The total number of optimised parameters in IT2 FLS employed triangular MFs is 25 if the FOU size is kept the same for both FSs, and 26 if the FOU size parameter is different for the two FSs in each input. Whereas, in case of Gaussian MFs, the total number of optimised parameters is 17 if the FOU size is kept the same for both FSs, and 18 if the FOU size parameter is different for the two FSs in each input. To compare, the total number of optimised parameters in IT2 FLS using 'conventional' FOU creation (SB) around the principal triangular T1 MFs is 49 if the FOU size is kept the same for both FSs and 50 if the FOU size parameter is different for the two FSs in each input. Also, the total number of optimised parameters in IT2 FLS using 'conventional' FOU creation (SB) around the principal Gaussian T1 MFs is 25 if the FOU size is kept the same for both FSs and 26 if the FOU size parameter is different for the two FSs in each input. The 'conventional' FOU creation method (i.e., standard blurring) has to define three (two) parameters in the case of a triangular (Gaussian) MF for the UMF and the same number for the LMF as well as the FOU size that is used to scale the LMF to form the FOU.

The training data is corrupted by zero-mean uniform noise. Three different signal-to-noise ratios (SNRs) are considered: 20, 10 and 0 dB. The noise-free (NF) data are also used. In each case, the IT2 FLS is learned and tuned on 500 points. The result for each case (triangular and Gaussian MFs) is four different tuned IT2 FLSs with the same FOU selection and another four tuned IT2 FLSs with different FOU selections at each noise level. The experiments are carried out and the results are shown in Figures 6.7 and 6.9 for the same FOU case for triangular and Gaussian MFs respectively and Figures 6.8 and 6.10 for the different FOU case for triangular and Gaussian MFs respectively.

After finishing the design of the IT2 FLSs with the resulting FOU sizes for both cases, the testing data sets are used to test the performance of the individual IT2 FLSs when faced with different uncertainty/noise levels. The testing and evaluation of these systems will be presented next.

6.3.4 Fuzzy Logic Systems Evaluation

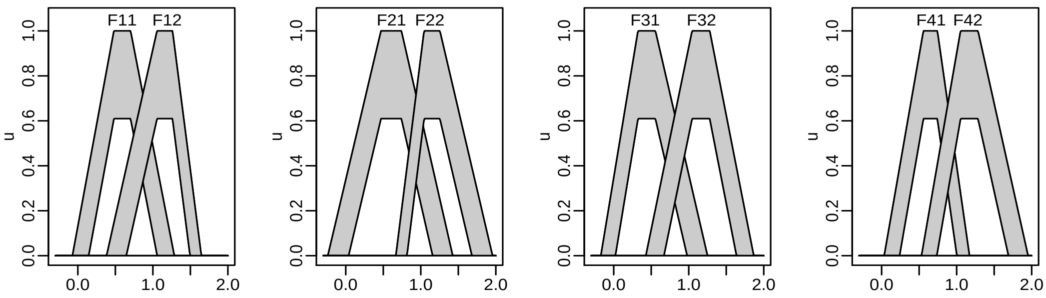
We test each of the eight IT2 FLSs with triangular MFS and the eight IT2 FLSs with Gaussian MFS resulting from the SA learning and tuning against four different noise levels with the two different testing sets. Each test is repeated 30 times to account for the random generation of the uniform noise. The performances of all the FLSs were evaluated using the RMSE for the testing data set 1:

$$RMSE = \sqrt{\frac{1}{200}\sum_{t=1508}^{1707}\left[s\left(t+1\right)-f\left(\mathbf{s}^{(t)}\right)\right]^2} \tag{6.1}$$

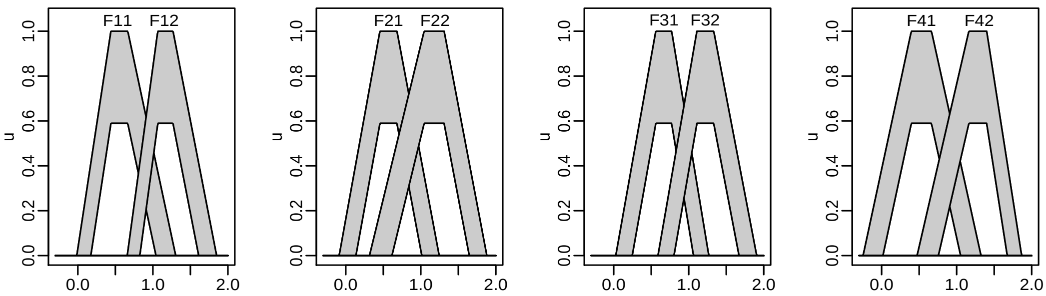
and the performances of all the FLSs were also evaluated using the RMSE for the testing data set 2:

$$RMSE = \sqrt{\frac{1}{200}\sum_{t=1508}^{2503}\left[s\left(t+1\right)-f\left(\mathbf{s}^{(t)}\right)\right]^2} \tag{6.2}$$

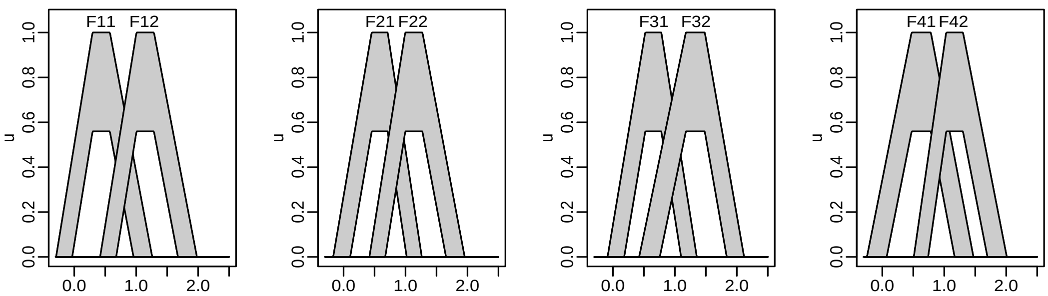
where, $s(t+1)$ is the output of the noisy testing data and $f(s^{(t)})$ is the crisp output of the FLS, and, $\mathbf{s}^{(t)} = [s(t-3), s(t-2), s(t-1), s(t)]^T$.



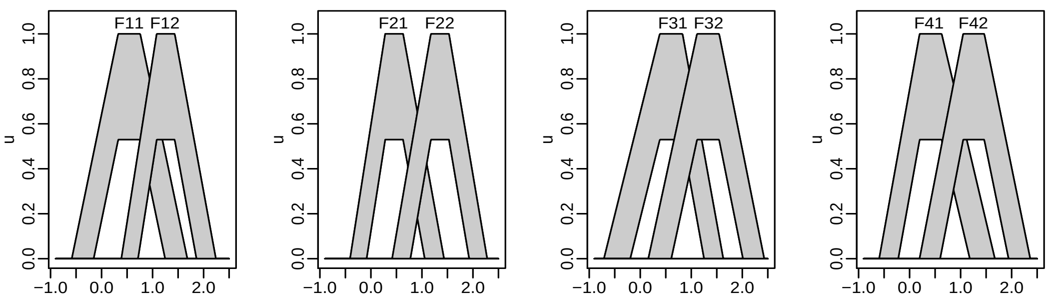
(a)



(b)

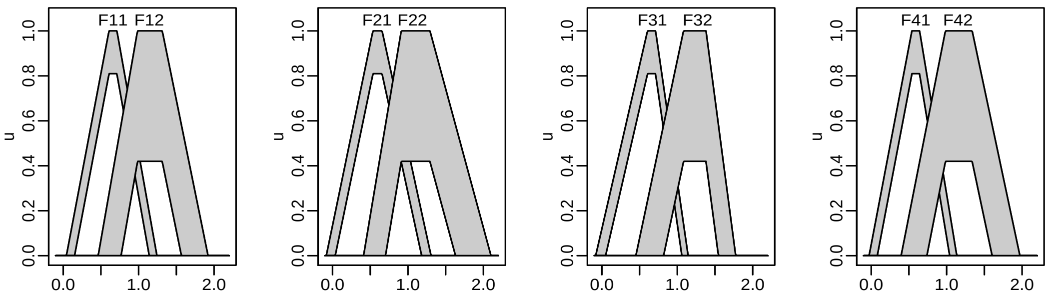


(c)

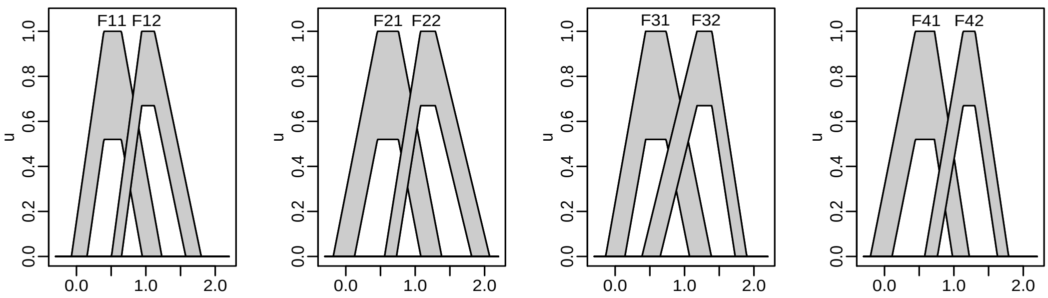


(d)

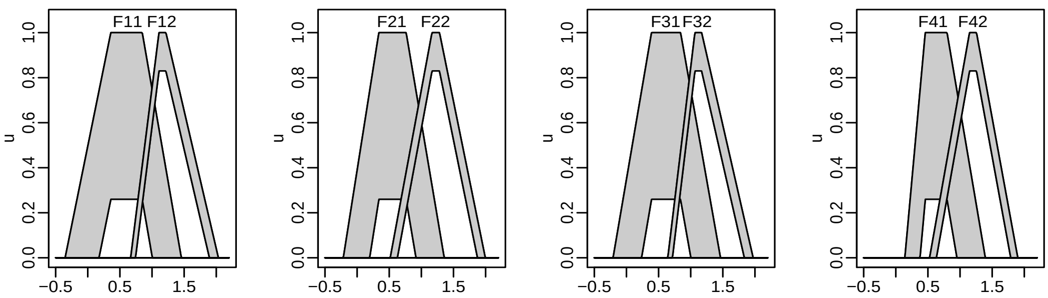
Figure 6.7 The resulting FOU (the same FOU size for both FSs) for the 4-inputs of the IT2 FLS using triangular MF trained with (a) a NF training set and trained with training sets corrupted with a uniform noise at different SNR values: (b) 20dB, (c) 10dB and (d) 0dB.



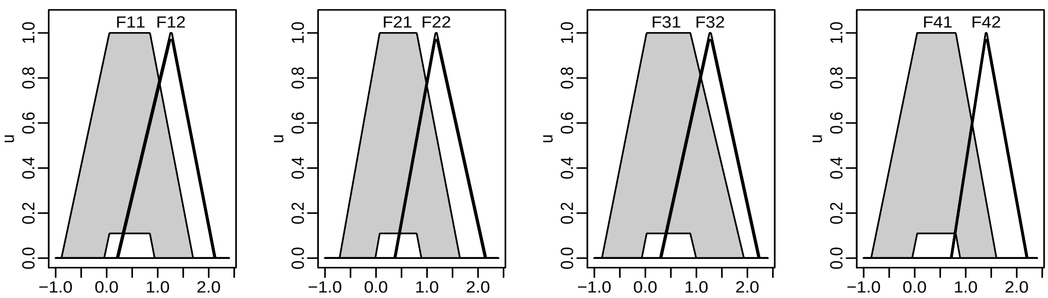
(a)



(b)



(c)



(d)

Figure 6.8 The resulting FOU (different FOU size for each FS) for the 4-inputs of the IT2 FLS using triangular MF trained with (a) a NF training set, and trained with training sets corrupted with a uniform noise at different SNR values: (b) 20dB, (c) 10dB and (d) 0dB.

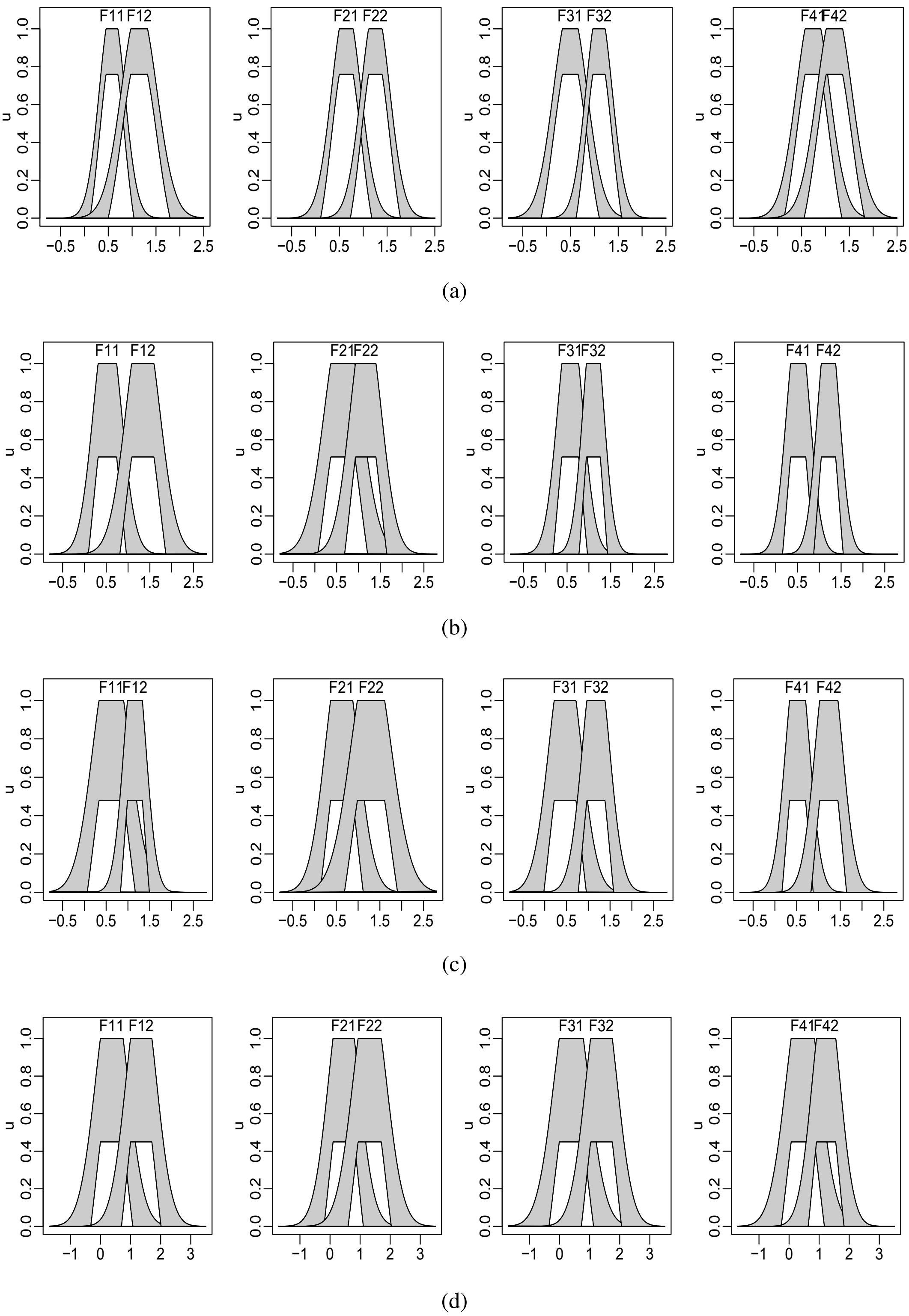


Figure 6.9 The resulting FOU (the same FOU size for both FSs) for the 4-inputs of the IT2 FLS using Gaussian MF trained with (a) a NF training set and trained with training sets corrupted with a uniform noise at different SNR values: (b) 20dB, (c) 10dB and (d) 0dB.

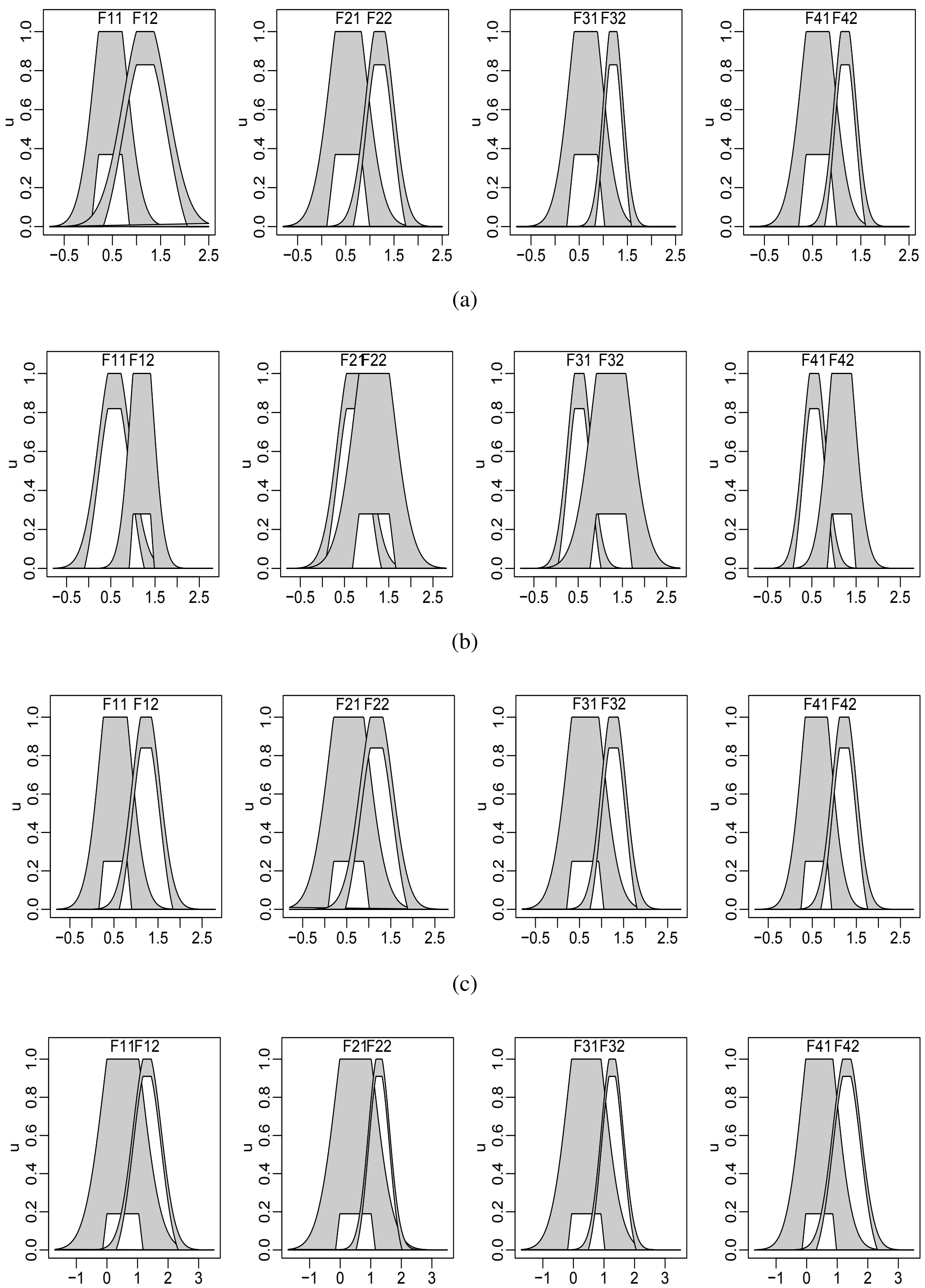


Figure 6.10 The resulting FOU (different FOU size for each FS) for the 4-inputs of the IT2 FLS using Gaussian MF trained with (a) a NF training set, and trained with training sets corrupted with a uniform noise at different SNR values: (b) 20dB, (c) 10dB and (d) 0dB.

Table 6.1 Average RMSE results of testing the optimised FLSs with triangular MF on different noise levels.

The training noise level	FOU selection case	FOU size c_1 for FS 1	FOU size c_2 for FS 2	Testing Data Sets 1 Results				Testing Data Sets 2 Results			
				NF	20	10	0	NF	20	10	0
NF	Same	0.39	0.39	0.10034	0.10221	0.11219	-	0.09580	0.09720	0.10653	-
	Different	0.19	0.58	0.10607	0.10664	0.11446	-	0.10171	0.10202	0.10851	-
20	Same	0.41	0.41	0.09637	0.09817	0.10953	-	0.09227	0.09391	0.10476	-
	Different	0.48	0.33	0.09679	0.09817	0.10919	-	0.09295	0.09410	0.10424	-
10	Same	0.44	0.44	0.09431	0.09597	0.10817	-	0.09112	0.09244	0.10402	-
	Different	0.74	0.17	0.09586	0.09679	0.10528	-	0.09229	0.09324	0.10129	-
0	Same	0.47	0.47	0.08304	0.08489	0.09747	0.17380	0.08198	0.08343	0.09498	0.16573
	Different	0.89	0.03	0.10055	0.10020	0.10278	0.16289	0.09515	0.09516	0.09845	0.15683

Table 6.2 Average RMSE results of testing the optimised FLSs with Gaussian MF on different noise levels.

The training noise level	FOU selection case	FOU size c_1 for FS 1	FOU size c_2 for FS 2	Testing Data Sets 1 Results				Testing Data Sets 2 Results			
				NF	20	10	0	NF	20	10	0
NF	Same	0.24	0.24	0.10431	0.106112	0.11514	0.19201	0.10015	0.10151	0.11058	0.18198
	Different	0.63	0.17	0.09722	0.09935	0.11246	0.19391	0.09378	0.09581	0.10813	0.18552
20	Same	0.49	0.49	0.10467	0.1532	0.11325	0.19188	0.10128	0.10169	0.10823	0.18438
	Different	0.18	0.72	0.09769	0.09900	0.11041	0.18881	0.09244	0.09383	0.10473	0.17995
10	Same	0.52	0.52	0.09831	0.09814	0.10447	0.19123	0.09502	0.09471	0.10080	0.18130
	Different	0.75	0.16	0.08914	0.09113	0.10368	0.18321	0.08551	0.08728	0.09969	0.17446
0	Same	0.55	0.55	0.08607	0.08804	0.10151	0.17089	0.08353	0.08531	0.09757	0.16341
	Different	0.81	0.09	0.09702	0.09727	0.10335	0.16488	0.09399	0.09411	0.09918	0.15641

The RMSE results are averaged over 30 runs and are depicted in Table 6.1 for triangular MF and in Table 6.2 for Gaussian MF, showing the results of the average RMSE of the MG time series tested using two different testing data sets (testing data set 1 and testing data set 2) as explained in Section 6.2.1 above.

6.4 Results and Discussion

In this section, the experiment results are analysed and discussed. We have conducted this study based on the CB method and using SA to optimise our designed systems and provide the 'best' FOU size.

From Tables 6.1 and 6.2, there are two FOU selection cases resulting from different learning and tuning processes at different noise levels (column 1). The FOU sizes that are selected on the basis of the SA algorithm are shown in columns 3 and 4.

The cells in Table 6.1 containing the symbol '–' indicate that no result was obtained due to the inability of the FLS predictor to work. This was probably because of the high level of noise at SNR of 0dB, indicating that these FLSs should be trained on lower levels of noise. However, the FLS that learned on the noise level of SNR equal to 0dB is working well.

From Table 6.1, it can be seen that as noise increases, the FOU size increases as well. From the results of the same FOU selection cases, FOU started with $c = 0.39$ at the noise-free (NF) training data up to $c = 0.47$ at the noise level with SNR = 0dB. The best performance achieved is shown at the training level of 0 db tested with NF data and the FOU size is 0.47. An example of the output results of the optimal selected FOU size tested using 10dB noise level is depicted in Figure 6.11.

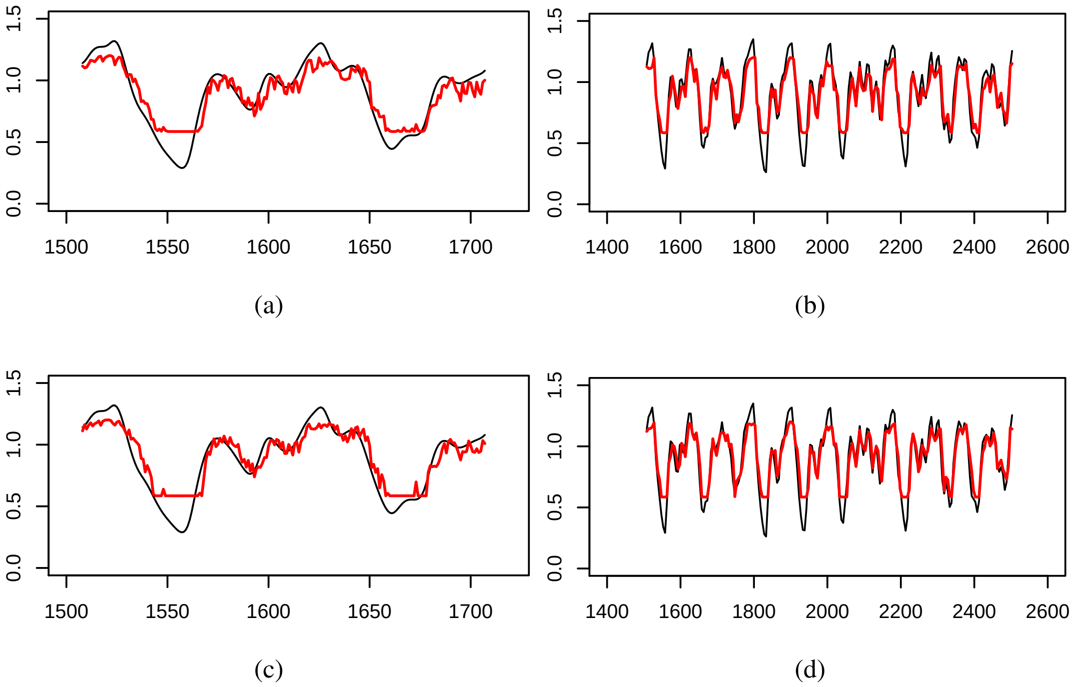
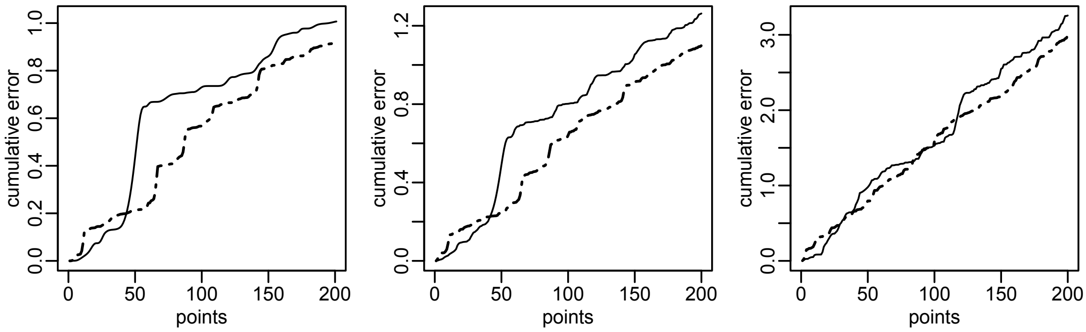


Figure 6.11 Sample output results of the optimal selected FOU size tested using 10dB noise level. (a) Same FOU size using testing data set 1. (b) Same FOU size using testing data set 2. (c) Different FOU size using testing data set 1. (d) Different FOU size using testing data set 2.

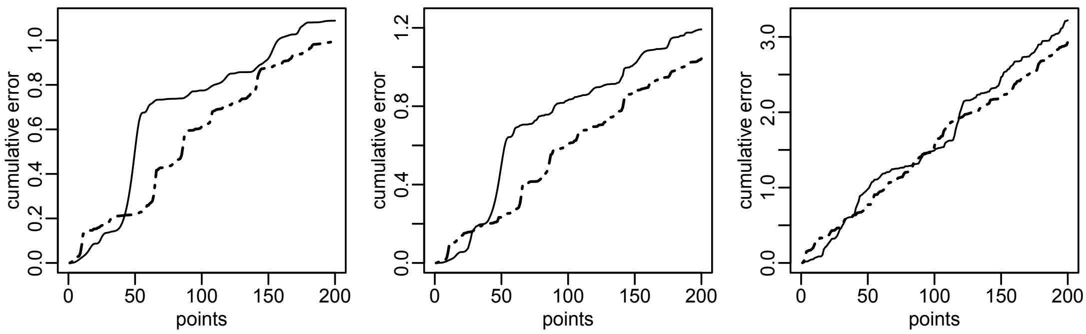
From Table 6.2, FOU started with $c = 0.24$ at the noise-free (NF) training data up to $c = 0.55$ at the noise level with SNR = 0dB. The best performance achieved is shown at the training level of 0 db tested with NF data and the FOU size is 0.55. From both MFs RMSE results, we can see that the Gaussian MF has a performance improvement over the triangular case and producing an output even with testing data that corrupted with high noise level (i.e., 0dB).

Generally, the best performance is achieved by the IT2 FLS that has the same FOU size for both sets, which was learned with the training data corrupted with noise level 0dB, giving the lowest RMSE of the two testing data sets.



(a) (b) (c)

Figure 6.12 The cumulative prediction error result from the two testing methods for triangular MF: testing data set1 (bold line) and testing set2 (dotted line) at different noise levels (a) NF, (b) 10dB and (c) 0dB.



(a) (b) (c)

Figure 6.13 The cumulative prediction error result from the two testing methods for Gaussian MF: testing data set1 (bold line) and testing set2 (dotted line) at different noise levels (a) NF, (b) 10dB and (c) 0dB.

The best results of testing the eight IT2 FLSs on different noise levels are achieved by the testing data set 2. Figures 6.12 and 6.13 for triangular and Gaussian MFs respectively, show the cumulative prediction error result from testing the FLSs on the two different testing sets. From both figures, we can see that the prediction errors are reduced by using the second testing approach.

6.5 Summary

Having established an strategy for appropriate FOU selection for a given system, the question of how we can further refine it in case the environment or the conditions change is an important one to ask. In this case, the FOU refinement is crucial to enable better adaptation in the new conditions. Thus, the objective of this chapter to refine the FSs and select the optimal FOU size in a given application faced with varying levels of uncertainty in order to enable a better adaptation in real world applications by utilising the optimisation of IT2 FLSs and hence, satisfying requirements 1, 2 and 4 stated in Chapter 3 (Section 3.2).

We proposed an optimised controlled blurring (O-CB) method used to optimise and select an optimal FOU size of an IT2 FLS. The CB method simplifies the process that is used to design/tune an IT2 FLS by using few parameters and using an SA algorithm to tune the shape of the FOUs. In this method, the IT2 FOU is basically defined by T1 MF parameters and one extra parameter that defines the FOU size. We then apply the well-known simulated annealing optimisation method, as this is an effective search method used in many fuzzy logic applications. The proposed approach is applied to the time series prediction problem given by the MG time series and using two different MFs: triangular and Gaussian.

Our approach consists of two cases for the FOU selection in addition to the learning and tuning applied to the other parameters of the FLSs. These cases were chosen to investigate the effect of the FOU size of different FSs on uncertainty capturing. The final IT2 FLSs with learned and tuned parameters were tested with the two different testing data sets at different levels of noise, showing a reduction in the cumulative prediction error. The FOU design mechanism, in conjunction with the SA optimisation method, provides an appropriate FOU selection process that produces IT2 FLSs that are able to handle highly uncertain data. Hence, the fourth objective declared in Chapter 1 is met.

In the next chapter, we present an approach to systematically give meaning (through a model) to the commonly used linguistic terms, 'low', 'medium', and 'high' uncertainty captured by the SNR.

Chapter 7

Linguistic Naming of the Uncertainty Levels

7.1 Introduction

The question 'What is high uncertainty?' is extremely frequent in fuzzy logic and particularly the type-2 fuzzy logic community–and for good reason. Further, as the level of uncertainty within a given application is the main driver in terms of the design decision between the use of type-1 and type-2 FLSs (and further, for the definition of the FOU size in the latter case), quantifications of what 'low', 'medium' and 'high' levels of uncertainty really mean are needed.

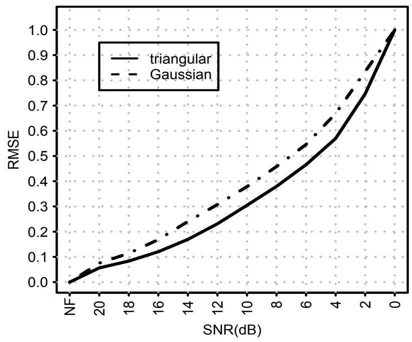
As outlined in the introduction, in the existing literature, in particular in the areas of T2 FLSs, uncertainty levels are frequently not quantified beyond a vague qualitative linguistic assessment such as 'high uncertainty'. Further, where such qualitative linguistic assessments are used, it is commonly unclear what they mean, i.e., what level of measurable uncertainty they refer to. Focussing on the explored SNR context, in this chapter, we provide an initial approach to develop fuzzy set based models that quantify these linguistic terms.

In engineering applications, the level of noise is commonly measured by SNR, where a high SNR refers to a clear signal (low amount of noise) and a low SNR refers to a noisy signal (high amount of noise). The universe of discourse used to describe uncertainty in FLSs is often captured using the linguistic labels ('low', 'medium' and 'high') and associated FSs. So, the quantification of uncertainty based on relating levels of the well-established SNR to the linguistic terms will inform and define what 'low', 'medium' and 'high' uncertainty means in an SNR context.

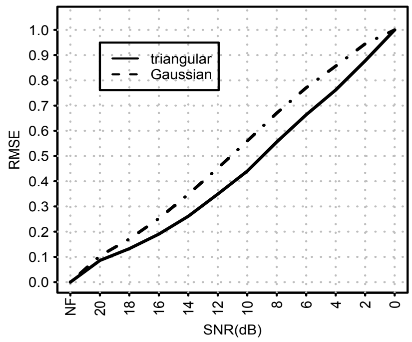
In this chapter, we use the results from Chapter 4 and provide an initial FS-based quantification (and underlying methodology) for the linguistic terms (i.e., 'low', 'medium' and 'high') commonly used in the fuzzy systems field to describe applications. While this quantification is application-context dependent, it provides a first step toward defining what we mean by, for example, a 'high' level of uncertainty in a given application.

7.2 General Model for Linguistic Uncertainty Quantification

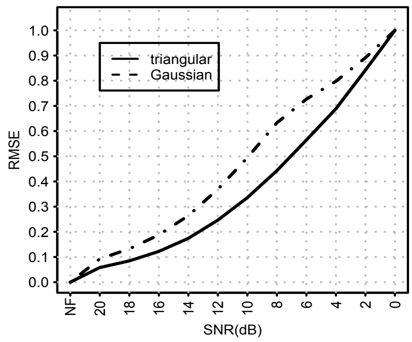
In this section, we present an approach to systematically assign meaning (through a model) to commonly used linguistic terms such as 'low', 'medium' and 'high' uncertainty. While our approach has been developed in the context of SNR and TSP, we believe it provides a valuable step toward establishing more precisely what is meant by 'low', 'medium' and 'high' uncertainty in FLS applications.



(a)



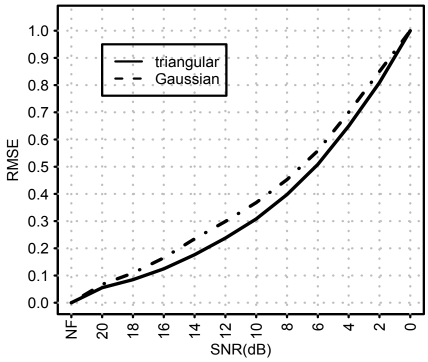
(b)



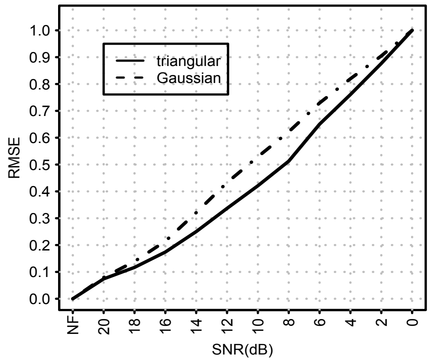
(c)

Figure 7.1 Normalized minimum RMSE values of uniform noise for triangular FSs (bold line) and for Gaussian FSs (dotted line) for (a) the MG, (b) IK and (c) LZ time series.

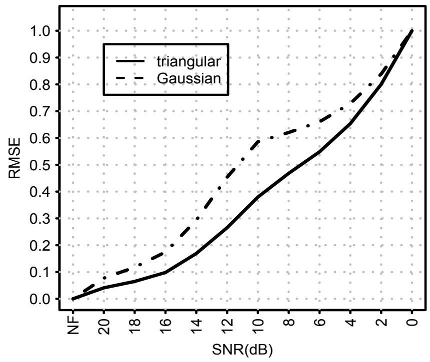
From the RMSE results in Chapter 4 shown in Tables 4.1-4.6, we consider the MG, IK and LZ time series results to demonstrate the modelling approach. We use the average RMSE values obtained for the three time series for each case (triangular and Gaussian) by taking the minimum value (bold values in the tables) at each noise level (SNR), and we normalize them in [0,1] for better representation. We proceed to use the normalized RMSEs as a guide to establish quantifications for individual linguistic noise level 'regions', e.g., 'low', 'medium' or 'high'.



(a)



(b)



(c)

Figure 7.2 Normalized minimum RMSE values of Gaussian noise for triangular FSs (bold line) and for the Gaussian FSs (dotted line), for (a) the MG, (b) IK and (c) LZ time series.

Figure 7.1 (for uniform noise) and Figure 7.2 (for Gaussian noise) show the normalized RMSE for the MG, IK and LZ time series (both triangular and Gaussian cases), where the behaviour of the RMSE over varying SNRs is shown.

In order to generate models (FSs) that quantify a specified number of (in our case three) linguistic labels describing levels of uncertainty, we follow the three-step process given below. We provide a detailed example by taking Figure 7.1(a) and then we explain the process of creating the three example, piecewise FSs as shown in Figure 7.3 for low, medium and high levels of uncertainty. The data that is used to construct the FSs in Figure 7.3 are shown in Table 7.1, 7.2 and 7.3 for low, medium and high uncertainty respectively. The details of the steps follow.

7.2.1 Step 1: Define the Uncertainty Regions

Divide the y-axis domain [0,1] as shown in Figure 7.3 into three overlapping subregions: the region [0,0.5] as the 'low' uncertainty/noise level (Figure 7.3(a) and Table 7.1), the region [0.25,0.75] as the 'medium' uncertainty/noise level (Figure 7.3(b) and Table 7.2) and the region [0.5,1] as the 'high' uncertainty/noise level (Figure 7.3(c) and Table 7.3). Each region captures a segment of the curve and in turn will represent a certain range of the SNRs shown on the x-axis. The selection of these regions is clearly subjective; we advocate the regular split of the axis used here as a reasonable approach.

7.2.2 Step 2: Define the Membership Functions

Each subregion is employed to define the MF of the respective FS and thus linguistic label. As such, for shoulder FSs such as for *low* and *high* in our case, the endpoints of the intervals are associated with the memberships 1 and 0 for left-shoulder FSs, and 0 and 1 for right-shoulder FSs. For non-shoulder FSs, we assume a central peak, and thus associate the interval endpoints with a membership of 0, while the interval midpoint is associated with a membership of 1.

7.2.3 Step 3: Normalise and Assign Membership Value

Finally, the values of each region are normalised to [0,1] as shown in Tables 7.1-7.3 and transposed to give rise to the actual membership functions between the established

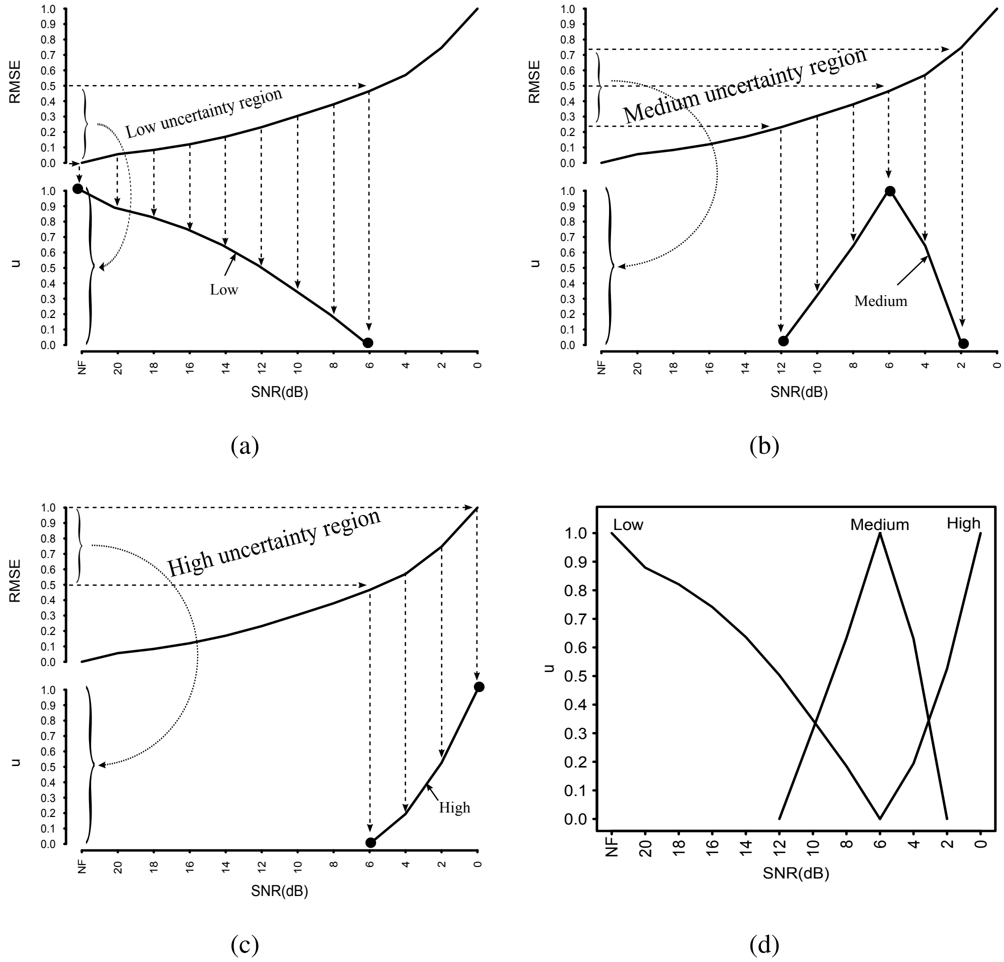


Figure 7.3 An illustration of the steps of uncertainty quantification of the noise levels for the MG TSP (triangular case). (a) Steps to find the low level, (b) the medium level and (c) the high level of uncertainty and (d) the linguistic quantification of the SNR for the MG TSP (triangular case). The bold points in (a), (b) and (c) are the endpoints.

Table 7.1 Detail of data underpinning Figure 7.3(a).

Noise level (x-axis)	‘low’ [0-0.5]	normalised RMSE	1 - normalised RMSE (y-axis)
NF	0	0	1
20dB	0.0563	0.1209	0.8791
18dB	0.0834	0.1791	0.8209
16dB	0.1205	0.2586	0.7414
14dB	0.1694	0.3637	0.6363
12dB	0.2314	0.4968	0.5032
10dB	0.3046	0.6540	0.3460
8dB	0.3797	0.8152	0.1848
6dB	0.4657	1.0	0.0

endpoints as shown in Figure 7.3. Where applicable (as for the FS ‘low" and the right half of the MF of the FS ‘medium"), we employ the complement as shown in Table 7.1 (4^{th} column) and Table 7.3 (6^{th} column) to establish the membership function. The resulting FSs for the linguistic quantification of the SNR for this example are shown in 7.3(d)

Examples for the assignments of uncertainty levels (SNRs) for each time series result for both MF types (triangular and Gaussian) are shown in Figure 7.4 for uniform noise and Figure 7.5 for Gaussian noise.

Table 7.2 Detail of data underpinning Figure 7.3(b).

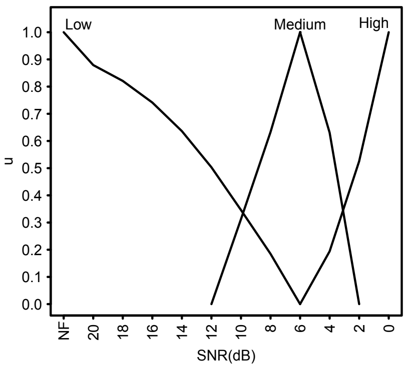
Noise level (x-axis)	'medium' left shoulder [0.25,0.5]	normalised RMSE (y-axis)	'medium' right shoulder [0.5,0.75]	normalised RMSE	1 - normalised RMSE (y-axis)
12dB	0.2314	0.0	–	–	–
10dB	0.3046	0.3124	–	–	–
8dB	0.3797	0.6328	–	–	–
6dB	0.4657	1.0	0.4657	0.0	1.0
4dB	–	–	0.5695	0.3694	0.6306
2dB	–	–	0.7465	1.0	0.0

Table 7.3 Detail of data underpinning Figure 7.3(c).

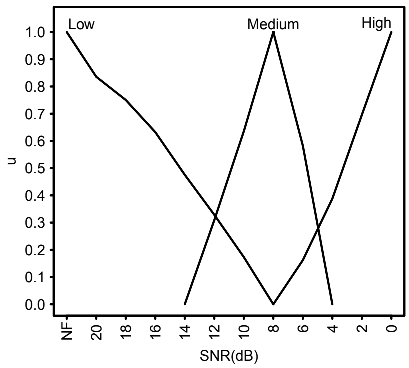
Noise level (x-axis)	'high' [0.5,1.0]	normalised RMSE (y-axis)
6dB	0.4657	0.0
4dB	0.5695	0.1942
2dB	0.7465	0.5256
0dB	1.0	1.0

7.3 Analysis and Discussion

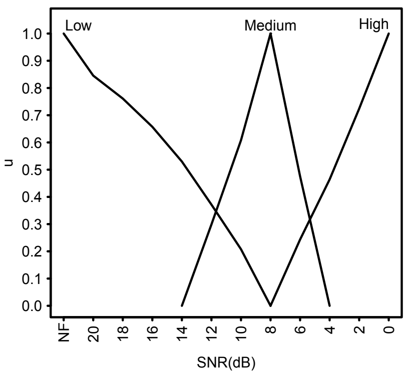
To show the assignments of uncertainty levels (SNRs) for each time series result for both cases (triangular and Gaussian), we follow the steps described in Section 7.2 above. The resulting FSs for the individual time series and MF types used in this thesis are depicted in Figure 7.4 for uniform noise and Figure 7.5 for Gaussian noise, in all cases, generating three FSs, for low, medium and high. These steps are also applicable to any number of chosen FSs. For example, Figure 7.6 shows a five FS example with



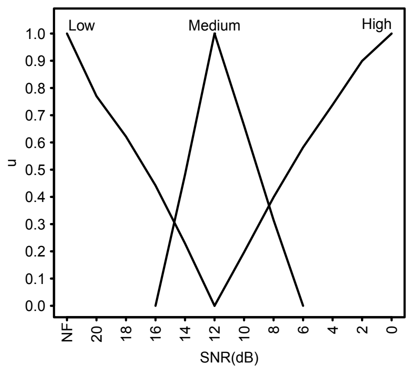
(a)



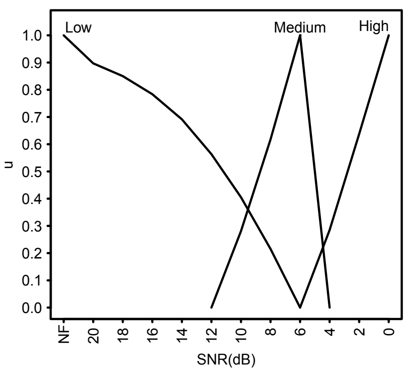
(b)



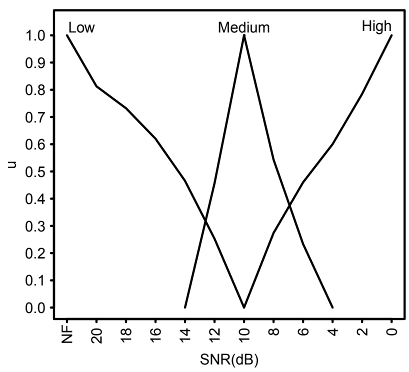
(c)



(d)



(e)



(f)

Figure 7.4 Uncertainty quantification of the noise levels for the MG, IK and LZ time series using uniform noise. (a,b), (c,d) and (e,f), show the linguistic quantification of the SNR for all three time series and for triangular and Gaussian FSs respectively.

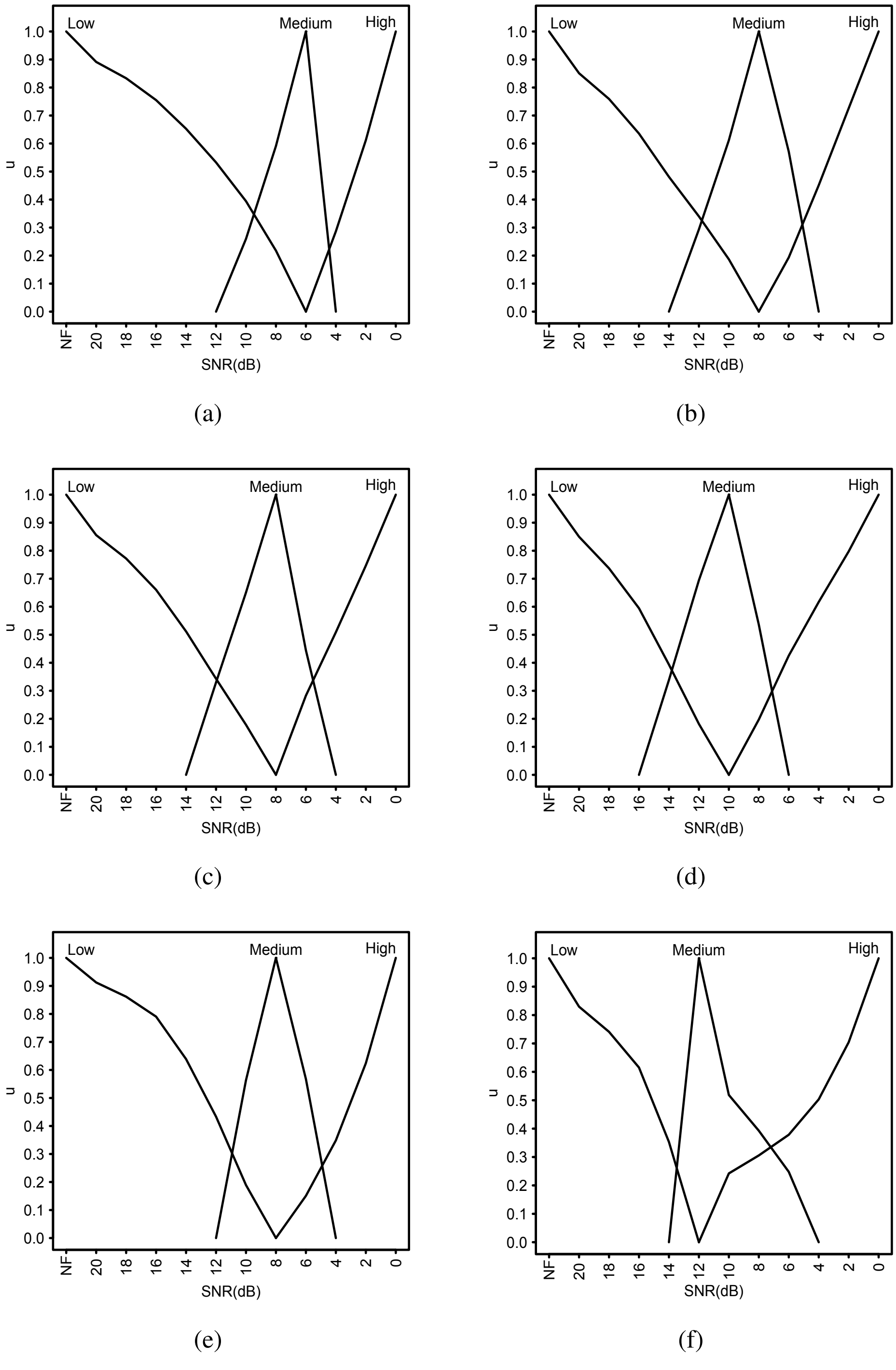


Figure 7.5 Uncertainty quantification of the noise levels for the MG, IK and LZ time series using Gaussian noise. (a,b), (c,d) and (e,f), show the linguistic quantification of the SNR for all three time series and for triangular and Gaussian FSs respectively.

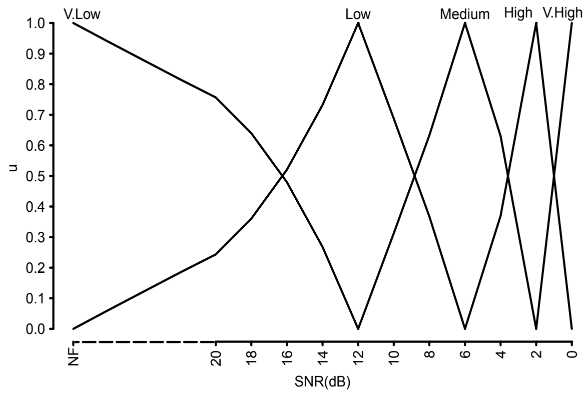


Figure 7.6 Uncertainty quantification of the noise levels for the MG triangular case using 5 FSs.

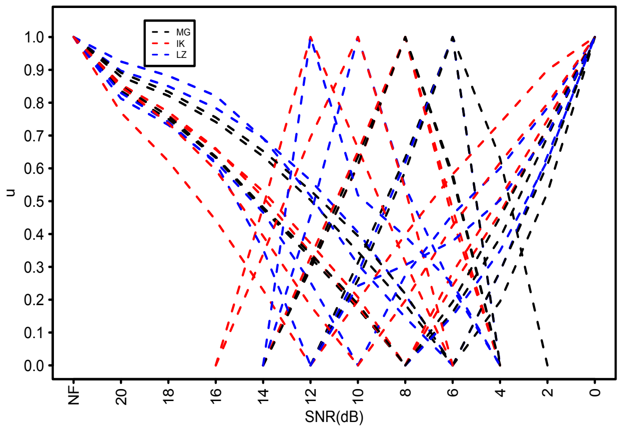
the linguistic labels (‘very low’, ‘low’, ‘medium’, ‘high’ and ‘very high’) applied to the uncertainty of the MG time series (triangular MFs case with uniform noise).

To the best of our knowledge, this proposed quantification of linguistic uncertainty levels is the first such attempt to provide models that quantify the frequently used labels of ‘low’, ‘medium’ and ‘high’ applied to uncertainty. These models provide a formalisation of the nature of the uncertainty/noise levels of a particular application and provide a framework for the generalisation to a domain of applications. Specifically, while in this case the resulting sets are designed directly based on a TSP application and an SNR context, generalisation of the resulting models for application areas (e.g., time series prediction generally) seems possible.

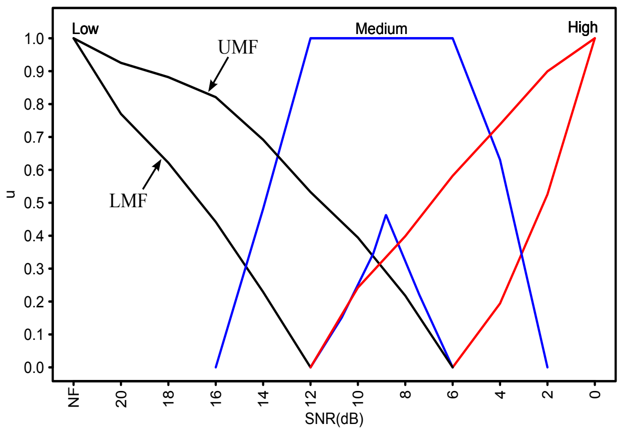
In this regard, the use of T2 FSs to capture collections of individual (T1) models [113] and thus to enable generalised quantifications of uncertainty seems a viable option. The collection operation or ‘collection operator’ proposed in [113] was developed to convert multiple T1 FSs into a single IT2 FS or GT2 FS. In this work we consider only transforming multiple T1 FSs to a single IT2 FS. For more details on this concept, we refer the reader to [113]. As an initial step toward this, adopting the proposed approach in [113], in Figure 7.7, we show an IT2 FSs which capture the individual T1 FSs. Specifically, Figure 7.7(a) shows the collection of the individual T1 FSs taken from the results shown in Figure 7.4 and Figure 7.5 for the linguistic quantification of the SNR, for the MG, IK and LZ time series (triangular and Gaussian FSs) and for the noise types (uniform and Gaussian). After applying the collection operation [113] to these FSs, the IT2 FSs resulting from this operation are depicted in Figure 7.7(b).

Importantly, these quantifications (models) of uncertainty allow us to summarize what ‘low’, ‘medium’ and ‘high’ levels of uncertainty mean in regards to the SNR in the context of the TSPs reviewed in this paper. In turn, we believe that the approach more generally is useful for the research community to empirically build up a picture of standard categorisation of uncertainty levels in different application categories (e.g., general time series prediction, control applications, computing with words, etc.) over time. Naturally, the proposed quantifications of commonly used linguistic labels is not a “solution” to what the linguistic terms mean to all individuals, however, it does provide a substantial step towards quantifying this meaning and leverages the degrees of freedom of IT2 FS to also capture uncertainty in the models.

Figure 7.7 allowing us to know in advance what the linguistic level of uncertainty (e.g., ‘low’, ‘medium’ or ‘high’) is, based on the application. For the example of an SNR of 10dB, if we enquire what the linguistic level of uncertainty is, from Figure 7.7(b), we see that 10db is a ‘medium’ uncertainty level in the general TSP applications adopted in this thesis. Conversely, Figure 7.7(b) also allows us to see what ‘low’, ‘medium’ and ‘high’ levels of uncertainty mean in regards to the SNR in the context of the TSPs reviewed in this thesis.



(a)



(b)

Figure 7.7 IT2 FSs collection operation. (a) Individual T1 FSs taken from Figure 7.4 and Figure 7.5. (b) IT2 FSs collection result that generated from the collection of the T1 FSs in (a).

7.4 Summary

In this chapter, the third objective stated in Chapter 1 is fulfilled by providing an approach to develop FS-based models that quantify linguistic terms such as ‘low’, ‘medium’ and ‘high’ uncertainty in the context of TSP.

We used the results provided in Chapter 4 and provide an initial FS-based quantification (and underlying methodology) for the linguistic terms commonly used in the fuzzy systems field to describe application specific uncertainty levels (e.g., ‘low’, ‘medium’ and ‘high’). Again, while this quantification is application-context dependent, it provides a first step toward defining what we mean when we argue that there is, for example, a ‘high’ level of uncertainty in a given application.

In the next chapter, the conclusions and contributions of this thesis are presented including limitations and the proposed direction for future work.

Chapter 8

Conclusions

This Chapter concludes the thesis by summarising the key points, outcomes and limitations of the research and discussing directions for future work.

The aim and research question of this thesis was stated in Chapter 1 and is repeated here. The main aim is to answer the following research question: 'What is the relationship between the size of the FOU of an IT2 FS and the uncertainty levels in a given application and how can understanding this relationship inform the design of FLSs?'

The results of this thesis show that there is a direct relationship between the FOU size of the FS and the uncertainty/noise levels. This research develops several approaches (see Chapter 3) to address the thesis aim and study the relationship between the FOU size of T2 FSs and the level of uncertainty. The question of 'what is high uncertainty?' and related questions such as 'When are T2 FLSs viable in comparison to T1 FLSs?' or, 'How much uncertainty warrants the use of T2 FLSs?' and exactly 'How "wide" should the FOUs of the respective T2 FSs be?' are interrelated sub-questions that are also addressed in this thesis. From the results of Chapters 4, 5 and 6, we can conclude that as the noise level increases, the best results (minimum RMSE) occur as the FOU size increases. We highlight that the choice between T1 FLS and IT2 FLS is continuous, i.e. from no FOU (T1) to very wide FOUs (ITSs). In other words, as the amount of uncertainty increases, the FOU of the FSs grows from initially T1 FSs to wider and wider IT2 FSs. This validates our intuition that higher uncertainty needs wider FOUs. Based on the experimental results in Chapter 4, we also provide the results of the optimal FOU size for specific TSP experiments followed by more elaborated results in the context of a concise set of requirements for the transitions from T1 to IT2 FS, providing initial guidelines on FOU size selection based on aggregate results across all experiments conducted.

We demonstrated and analysed the performance of a range of carefully designed FLSs in the context of the time series analysis. This is done by following the FLS design and evaluation process presented in Chapter 3 which provides a general framework to enable a systematic design and transition from T1 FLSs to IT2 FLSs and to test their results. Through contrasting the performance of these resulting FLSs in the face of inputs with varying signal-to-noise ratios (SNRs) in a rich set of time-series prediction (TSP) experiments (investigated in Chapters 4, 5 and Chapter 6), the distinct pattern of performance arising from the different levels-of-uncertainty and FOU-size combinations is explored and captured.

In Chapter 6 we applied and showed the results of an FOU refinement mechanism, in conjunction with the simulated annealing (SA) optimisation method, providing an efficient FOU selection process that produces IT2 FLSs able to handle varying uncertainty levels.

In Chapter 7, we developed FS-based quantifications of the linguistic labels of ‘low’, ‘medium’ and ‘high’ for the given application context, and highlighted the potential of such quantifications in giving meaning to linguistic uncertainty labels, commonly used in FLS research.

In this chapter, the contributions and the key findings of this thesis are summarised. The limitations of this research and the possible directions for further research are also discussed. Finally, a list of publications already produced as part of the work in this thesis are presented.

8.1 Contributions

To reach the aim and fulfil the objectives stated in Chapter 1, this thesis makes the following contributions:

8.1.1 The Uncertainty Indicator

The objective of the uncertainty indicator is to study the uncertainty modelling of IT2 FSs. The uncertainty indicator is intended to show the amount of uncertainty captured by FSs and modelled by their FOUs. Specifically, the uncertainty indicator captures the size of the primary membership interval at a given x for a given IT2 FS.

The uncertainty indicator provides a convenient mechanism to assess and compare the uncertainty inherent in IT2 FS memberships.

8.1.2 Controlled Blurring Method

The objective of this method is to relate the footprint of uncertainty (FOU) to the uncertainty level and vice versa. This novel method was introduced in Chapter 3 as apart of our investigative approach and then detailed in Chapter 4. The controlled blurring of T1 MFs to IT2 MFs ensures that the FOU size within a given IT2 fuzzy set remains constant (within the support of the lower membership function) thus addressing requirements 1 and 2 from Chapter 3 (Section 3.2). The latter is crucial if we seek to establish a direct link between a known level of uncertainty affecting a given variable and the FOU size of the fuzzy sets describing that variable.

Using the controlled blurring (CB) method, we investigate the relationship between the FOU size of T2 FSs and the level of presented uncertainty. In Chapter 4, we demonstrated and analysed the performance of a range of carefully designed FLSs in the context of the Mackey Glass, Ikeda and Lorenz time series. They are tested over different noise levels using both uniform and Gaussian noise and their results for Gaussian noise are consistent with the results of uniform noise. Both lead to the same observations and conclusions detailed further in Chapter 4. A direct relationship between the FOU sizes of the FSs and the noise levels was shown and illustrated, namely, as the noise level increases, the FOU that gives the minimum RMSE value increases as well. Also, it was observed that the T1 FLSs ($c = 0$) outperform IT2 FLSs on NF-testing data as intuitively expected.

In Chapter 5, this method was also implemented to design a series of IT2 FLSs tested under different conditions (Noise free and noisy training data, singleton and non-singleton fuzzification) with a detailed performance comparison and evaluation with the other proposed shape-preserving controlled blurring (SP-CB) method in the context of the MG time series. The results indicate that in general, FLSs based on CB outperform those based on SP-CB.

In Chapter 6, this method was again implemented as part of the investigation to design IT2 FLSs based on this approach (i.e., CB) and using simulated annealing optimisation to learn and tune the FLS parameters and select the optimal FOU size. The results show that this approach is an efficient FOU selection mechanism that produces IT2 FLSs with good performance in face of uncertain environment.

8.1.3 Shape-Preserving Controlled Blurring Method

The aim of the shape-preserving controlled blurring (SP-CB) method is to study and present an IT2 FS creation method. The SP-CB method is presented to systematically captures a specified amount of uncertainty (i.e., the uncertainty in memberships over the support of the LMF is constant) and preserves the original shape of the MF (the LMF and UMF keep their original T1 MF shape) for comparison between T1 and IT2 FLSs. Thus, SP-CB designed to address requirements 1, 2 and 3 from Chapter 3 (Section 3.2). In Chapter 3, as an alternative to the controlled blurring (CB) method, we proposed the shape-preserving controlled blurring (SP-CB) method to enable the transition of T1 to IT2 FSs for given 'levels' of uncertainty. Specifically, we propose to transition from T1 to IT2 FLSs through varying the size of the FOU of their respective FSs while maintaining the original FS shape and keeping the size of the FOU over the FSs as constant as possible.

The proposed approach (SP-CB) enables both the adaptation of the IT2 FS for known levels of uncertainty (i.e. by increasing FOU size with increasing uncertainty) and systematic comparison of the original T1 FLS(s) to the resulting new IT2 FLS(s).

This method is compared with the IT2 FS creation method CB, which follows a similar creation approach as SP-CB but does not maintain the MF shape.

In Chapter 5, in order to assess the viability and explore the behaviour of both SP-CB and CB, we conducted a detailed performance comparison and evaluation in the context of Mackey-Glass (MG) time series analysis. Both methods are tested under different conditions (Noise free and noisy training data, singleton and non-singleton fuzzification) as well as for triangular and trapezoidal MFs. The results indicate that, in general, FLSs based on CB outperform those based on SP-CB. However, both methods provide expected performance increases for increasing FOU sizes as uncertainty/noise levels increase.

Based on this, it seems that in applications where the systematic transition and comparison from an original T1 FLS is paramount, SP-CB is preferable, as it maintains the MF shapes. In applications where this level of comparability is not vital, however, CB provides superior levels of performance, while also maintaining a systematic and parametrised increase in FOU size in the face of increasing uncertainty. Finally, as expected, the non-singleton FLSs provide superior outputs to singleton FLSs and training the systems on noisy data produces better performance, especially with the singleton FLS cases.

8.1.4 Optimised Controlled Blurring Method

The objective of the optimised controlled blurring (O-CB) method is to select the optimal FOU size in a given application faced with varying levels of uncertainty in order to enable a better adaptation in real world applications by using an optimisation method. Building on the CB method, we provide a process to tune an IT2 FLS where the IT2 FOU is basically defined by T1 MF parameters and an extra parameter that defines the FOU size. We then apply the well-known simulated annealing (SA) optimisation method, an effective search method used in many fuzzy logic applications.

In Chapter 6, the proposed approach (O-CB) was applied to the time series prediction problem given by the MG time series. Our approach consists of two cases for the FOU selection in addition to the learning and tuning applied to the other parameters of the FLSs. These cases investigate the effect of the FOU size of different FSs on uncertainty capturing. The final IT2 FLSs with learned and tuned parameters are tested with the two different testing data sets at different levels of noise, showing a reduction of the cumulative prediction error. The results show that the proposed FOU design mechanism, in conjunction with the SA optimisation method, used in this work, provides an FOU selection process that produces IT2 FLSs that are able to handle highly uncertain data.

8.1.5 Determining FOU Sizes Based on Known Levels of Uncertainty

The objective of this aspect is to highlight and answer questions such as: ‘How much uncertainty warrants the use of T2 FLSs?’ and ‘Exactly how “wide” should the FOUs of the respective T2 FSs be?’

In Chapter 4, an analytical method that informs the selection of the FOU size in response to a known quantification of uncertainty for a given application was presented. This exploration approach showed how the level of uncertainty can be related to the size of the FOUs of FSs in order to ascertain the optimal size of FOUs for the actual level of uncertainty in a given application.

Based on the experimental results shown in Chapter 4, we first provide specific examples of the optimal FOU size for specific TSP experiments before providing initial guidelines on FOU size selection based on aggregate results across all experiments conducted. While it is clear that these guidelines are based on TSP experiments, they provide a valuable starting point for establishing more generically applicable guidelines for FOU size selection in applications where the level of uncertainty (e.g., SNR in our case) can be determined a priori.

8.1.6 Giving Meaning to Linguistic Uncertainty Labels

The main objective of this approach is to construct an FS-based quantification and modelling (and underlying methodology) for linguistic terms (e.g., 'low', 'medium' and 'high') commonly used in the fuzzy systems field to describe applications. While this quantification is application-context dependent, it provides a first step toward defining what is meant by the linguistic uncertainty labels as, for example, a 'high' level of uncertainty in a given application.

In Chapter 7 based on the results generated in Chapter 4, we provided a three-step approach to generate models (FSs) that quantify a specified number of linguistic labels describing levels of uncertainty. These steps are applicable to any number chosen FSs, for example, three FSs with the linguistic labels 'low', 'medium' and 'high', or five FSs with the linguistic labels 'very low', 'low', 'medium', 'high' and 'very high', all applied to the level of uncertainty, as presented in Chapter 7.

To the best of our knowledge, this proposed quantification of linguistic uncertainty levels is the first such attempt to provide models to quantify the frequently used labels of 'low', 'medium' and 'high' uncertainty. The models provide formalisation of the nature of the uncertainty/noise levels of a particular application and provide a framework for the generalisation to a domain of applications. Specifically, while in this case the resulting sets were designed directly based on a TSP application and an SNR context, generalisation of the resulting models for application areas (e.g., TSP generally) seems possible.

8.2 Limitations and Future Directions

In this section, in order to identify the limitations of this thesis and the suggested directions for future work, eight sub-sections are detailed below.

8.2.1 FOU Creation Methods in Real World Applications

This thesis provides a systematic empirical exploration of the behaviour of IT2 FLSs with different FOUs in the face of different levels of noise/uncertainty. We have proposed a carefully designed methodology for MF design which provides the underpinning for clear experimentation and analysis. A series of experiments is conducted using data sets generated from three well-known time series; Mackey Glass (in Chapters 4, 5 and 6), Lorenz (in Chapters 4 and 5) and Ikeda (in Chapter 4). These time series are particularly attractive for the evaluation of algorithms as all underlying parameters (ground truth, noise) can be controlled, something not possible in real-world control applications such as robotics. However, having focused on time series data, this thesis does not provide proof or fully generalizable evidence regarding the behaviour of all (IT2) FLSs or IT2 MFs, nor for all application areas. In order to reach generalisable conclusions, similar experiments should be conducted in the context of a wide range of real world applications, for example, time-series prediction applications such as financial forecasting and control applications. Further detailed and extensive studies should also be carried out into a range of commonly used MFs, including more ‘standard’ types of FSs.

8.2.2 Exploring General Type-2 Fuzzy Sets and Systems

Interval type-2 fuzzy logic sets and systems were extensively explored and studied in this thesis through a variety of approaches used to create the FOUs of the employed IT2 FSs. Looking further, general type-2 fuzzy sets and systems would be an interesting avenue for future work. The suggestion would be to explore the methodological generation of general type-2 fuzzy logic systems based on information on levels and distribution of uncertainty in given applications.

8.2.3 Rule Base Creation Method

In this work, the Wang-Mendel method (WM-method) for generating a rule base from numerical data was adopted. The WM-method is popular in the field of fuzzy systems because of its simplicity and effectiveness in data-driven fuzzy rule generation. The WM-method is a generic method for creating a fuzzy rule base that contains a combination of rules generated from numerical data and linguistic labels and often, associated MFs given by expert(s). Although its high performance has been demonstrated, it has a problem related to its way of selecting rules and the fact that their completeness is not guaranteed [175]. For the purpose of future work, we propose the investigation of approaches, such as the revised and extended Wang-Mendel method [176], or other approaches that take different learning algorithms as a base, such as neural networks (NNs), genetic algorithms (GAs) and clustering.

8.2.4 Optimisation Method

In Chapter 6, simulated annealing (SA) is used as a method of optimisation and an approach to selecting the 'best' FOU sizes in the employed IT2 FSs. SA is successfully used in the fuzzy system applications and more specifically in forecasting of Mackey-Glass time series (MG), as well as often requiring less CPU time than for example GAs. However, as a future avenue for research, other optimisation approaches such as NNs and GAs could be used with attention paid to controlling the computational time.

8.2.5 Modelling Uncertainty in the Input Fuzzy Sets Using Non-singleton Fuzzification

In this thesis, the modelling of uncertainty affecting the FLS inputs is conducted as part of the antecedent FSs. In principle however, established FLS methodology foresees the use of the input FSs for the modelling of input uncertainty while antecedent FSs model uncertainty in the linguistic antecedent terms. While in most singleton FLS applications, uncertainty in input and antecedent terms is "mixed" in the antecedent FSs, for the step-by-step modelling approach put forward in this thesis, exploring the application a non-singleton fuzzification approach would be highly valuable. In such an approach, the proposed methodology to determine an appropriate antecedent FS FOU would be adjusted to generate appropriate FOUs for the input FSs based on input uncertainty, achieving a more elegant and potentially more efficient FLS design.

8.2.6 Establishing the Inverse Process

In this thesis we have proposed a carefully designed methodology for FOU creation, which provides different FOUs in the face of different levels of noise/uncertainty. This can be considered as a 'forward' problem, i.e., going from a desired uncertainty level and parameter to an appropriate IT2 FS with associated centroid. However, an inverse approach may also be of interest, where a specific centroid is provided as measure of uncertainty and the method generates an appropriate IT2 FS for this centroid. As it is likely that for a given centroid a number of IT2 FSs could be generated, the result would probably take the form of design restrictions for the FSs, rather than a single IT2 FS.

8.2.7 Measuring Uncertainty in Fuzzy Sets

In this thesis, we introduced the uncertainty indicator which captures the amount of uncertainty associated with the degrees of membership of an IT2 fuzzy set (i.e. the size of the primary membership interval) resulting from the difference between the upper and lower membership value for a given x). While the centroid of an IT2 fuzzy set provides an indication of the uncertainty associated with the given IT2 FS as a whole, the proposed uncertainty indicator focuses specifically on the uncertainty in the primary membership arising from the interaction of a vertical slice at a specific x with a given IT2 MF. Both indicators of uncertainty provide useful yet very different information. In the future it would be of value to explore other or combined uncertainty indicators.

8.2.8 Uncertainty Quantifications

The quantifications (models) of uncertainty presented in this thesis allow us to summarise what ‘low’, ‘medium’ and ‘high’ levels of uncertainty mean in regards to the SNR in the context of the TSPs reviewed. In turn, we believe that the approach more generally is useful for the research community to empirically build up a picture of standard categorisations of uncertainty levels in different application categories (e.g., general time series prediction, control applications, computing with words, etc.) over time. Thus, a clear path for future work is the application of the proposed quantification methodology in a wider set of applications, providing further empirical examples and exploring the potential of the methodology more widely.

8.3 Publications Produced

All of the research presented in each of the thesis chapters has been published in leading international conferences and journals, we are looking to explore the future direction described in subsection 8.2.5 and to publish it in both an international conference and a journal. A full list of publications and presentations derived from this work follows.

8.3.1 Journal Publications

1. J. Aladi, C. Wagner, J. Garibaldi and A. Pourabdollah, ‘What is ‘High Uncertainty’ ? From Type-1 to Interval Type-2 FLSs Through Controlled Blurring,’ Manuscript submitted for publication in the Transaction of Fuzzy Systems Journal. Under review.

2. A. Pourabdollah, C. Wagner and J. Aladi, ‘Improved Uncertainty Capture for Non-Singleton Fuzzy Systems,’ Manuscript submitted for publication in the Transaction of Fuzzy Systems Journal. Accepted for publication in 2016.

8.3.2 Conference Publications

1. J. Aladi, C. Wagner, and J. Garibaldi, ‘Type-1 or interval type-2 fuzzy logic systems -on the relationship of the amount of uncertainty and FOU size,’ in Proceedings of the IEEE International Conference on Fuzzy Systems, 2014, pp. 2360–2367.

2. J. Aladi, C. Wagner, J. Garibaldi and A. Pourabdollah, ‘On Transitioning From Type-1 to Interval Type-2 Fuzzy Logic Systems,’ in Proceedings of the IEEE International Conference on Fuzzy Systems, 2015, pp. 1-8.

3. J. Aladi, C. Wagner, and J. Garibaldi, ‘A Simplified Method of FOU Design Utilising Simulated Annealing,’ in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Hong Kong, October 2015, pp. TBC.

4. A. Pourabdollah, C. Wagner and J. Aladi, ‘Changes under the Hood - a New Type of Non-Singleton Fuzzy Logic System,’ in Proceedings of the IEEE International Conference on Fuzzy Systems, 2015, pp. 1-8.

5. J. Aladi, C. Wagner, and J. Garibaldi, ‘An Investigation of Uncertainty Representation in Higher-Order Fuzzy Logic Systems,’ in Proceedings of the Eighth Saudi Students Conference in the UK, 2016, pp. 213-227.

8.3.3 Presentations

- Type-1 or Interval Type-2 Fuzzy Logic Systems -On the Relationship of the Amount of Uncertainty and FOU Size, The Intelligent Modelling and Analysis (IMA) Research Group seminar, School of Computer Science, The University of Nottingham, United Kingdom, January 13, 2014.

- An Investigation of Uncertainty Representation in Higher-Order Fuzzy Logic Systems, The 8th Saudi Students Conference in UK, the Queen Elizabeth II Conference Centre, London, United Kingdom, January 31, 2015.

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[2] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation," *Soft Computing*, vol. 1, no. 4, pp. 180–197, 1997.

[3] H. Hagras, "Type-2 FLCs: A new generation of fuzzy controllers," *IEEE Comput. Intell. Mag.*, vol. 2, no. 1, pp. 30–43, Feb 2007.

[4] R. John and S. Coupland, "Type-2 fuzzy logic: A historical view," *IEEE Comput. Intell. Mag.*, vol. 2, no. 1, pp. 57–62, 2007.

[5] C. Wagner and H. Hagras, "Toward general type-2 fuzzy logic systems based on zSlices," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 4, pp. 637–660, 2010.

[6] L. X. Wang, *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ, USA: Prentice-Hall, 1997.

[7] R. Yager and D. Filev, *Essentials of Fuzzy Modeling and Control*. New York, NY, USA: Wiley-Interscience, 1994.

[8] M. Versaci and F. C. Morabito, "Fuzzy time series approach for disruption prediction in Tokamak reactors," *IEEE Trans. Magn.*, vol. 39, no. 3, pp. 1503–1506, 2003.

[9] N. Karnik and J. Mendel, "Applications of type-2 fuzzy logic systems to forecasting of time-series," *Information Sciences*, vol. 120, no. 1, pp. 89–111, 1999.

[10] L. X. Wang and J. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, 1992.

[11] K. Hirota and W. Pedrycz, "Fuzzy computing for data mining," *Proc. IEEE*, vol. 87, no. 9, pp. 1575–1600, 1999.

[12] R. Yager, "Database discovery using fuzzy sets," *International Journal of Intelligent Systems*, vol. 11, no. 9, pp. 691–712, 1996.

[13] J. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[14] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning–I,” *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[15] D. Wu and J. Mendel, “Uncertainty measures for interval type-2 fuzzy sets,” *Information Sciences*, vol. 177, no. 23, pp. 5378–5393, 2007.

[16] J. Mendel, “Advances in type-2 fuzzy sets and systems,” *Information Sciences*, vol. 177, no. 1, pp. 84–110, 2007.

[17] ——, “General type-2 fuzzy logic systems made simple: A tutorial,” *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 5, pp. 1162–1182, Oct 2014.

[18] J. Mendel and R. John, “Type-2 fuzzy sets made simple,” *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 117–127, 2002.

[19] S. Coupland and R. John, “Geometric type-1 and type-2 fuzzy logic systems,” *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 1, pp. 3–15, 2007.

[20] J. Mendel and F. Liu, “On new quasi-type-2 fuzzy logic systems,” in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2008, pp. 354–360.

[21] F. Liu, “An efficient centroid type-reduction strategy for general type-2 fuzzy logic system,” *Information Sciences*, vol. 178, no. 9, pp. 2224–2236, 2008.

[22] L. A. Lucas, T. Centeno, and M. Delgado, “General type-2 fuzzy inference systems: analysis, design and computational aspects,” in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2007, pp. 1–6.

[23] S. Greenfield and R. John, “Optimised generalised type-2 join and meet operations,” in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2007, pp. 1–6.

[24] C. Wagner and H. Hagras, “zSlices–towards bridging the gap between interval and general type-2 fuzzy logic,” in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2008, pp. 489–497.

[25] N. Karnik and J. Mendel, “Operations on type-2 fuzzy sets,” *Fuzzy sets and systems*, vol. 122, no. 2, pp. 327–348, 2001.

[26] S. Greenfield, R. John, and S. Coupland, “A novel sampling method for type-2 defuzzification,” in *Proc. of UKCI 2005*, London, Sep. 2005, pp. 120–127.

[27] N. Karnik, J. Mendel, and Q. Liang, “Type-2 fuzzy logic systems,” *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 643–658, 1999.

[28] L. A. Zadeh, “Toward extended fuzzy logic–A first step,” *Fuzzy Sets and Systems*, vol. 160, no. 21, pp. 3175–3181, 2009.

[29] Q. Liang and J. Mendel, “Interval type-2 fuzzy logic systems: theory and design,” *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, 2000.

[30] J. Mendel and X. Liu, "Simplified interval type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 6, pp. 1056–1069, 2013.

[31] M. Schrieber and M. Biglarbegian, "Hardware implementation and performance comparison of interval type-2 fuzzy logic controllers for real-time applications," *Applied Soft Computing*, vol. 32, no. 0, pp. 175–188, 2015.

[32] T. Lin, H. Liu, and M. Kuo, "Direct adaptive interval type-2 fuzzy control of multivariable nonlinear systems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 3, pp. 420–430, 2009.

[33] C. Wagner and H. Hagras, "A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2007, pp. 1–6.

[34] N. Baklouti and A. Alimi, "Motion planning in dynamic and unknown environment using an interval type-2 TSK fuzzy logic controller," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2007, pp. 1–6.

[35] M. Biglarbegian, W. Melek, and J. Mendel, "On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling," *Information Sciences*, vol. 181, no. 7, pp. 1325–1347, 2011.

[36] O. Linda and M. Manic, "Interval type-2 fuzzy voter design for fault tolerant systems," *Information Sciences*, vol. 181, no. 14, pp. 2933–2950, 2011.

[37] ——, "Comparative analysis of type-1 and type-2 fuzzy control in context of learning behaviors for mobile robotics," in *36th Ann. Conf. on IEEE Ind. Electron. Soc., IECON 2010*, 2010, pp. 1092–1098.

[38] T. Dereli, A. Baykasoglu, K. Altun, A. Durmusoglu, and I. B. Turksen, "Industrial applications of type-2 fuzzy sets and systems: A concise review," *Computers in Industry*, vol. 62, no. 2, pp. 125–137, 2011.

[39] H. Hagras and C. Wagner, "Towards the wide spread use of type-2 fuzzy logic systems in real world applications," *IEEE Comput. Intell. Mag.*, vol. 7, no. 3, pp. 14–24, 2012.

[40] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. i," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404–418, 1990.

[41] M. Sugeno, "An introductory survey of fuzzy control," *Information Sciences*, vol. 36, no. 1, pp. 59–83, 1985.

[42] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, 2006.

[43] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Communications of the ACM*, vol. 37, no. 3, pp. 77–84, 1994.

[44] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[45] L. Holmblad and I. Ostergaard, "Control of a cement kiln by fuzzy logic," in *Fuzzy Information and Decision-Processes*, M. M. Gupta and E. Sanchez, Eds. Amsterdam, North-Holland, 1982, pp. 389–399.

[46] F. O. Karray and C. W. De Silva, *Soft computing and intelligent systems design: theory, tools, and applications*. Pearson Education, 2004.

[47] G. Klir and M. Wierman, *Uncertainty-Based Information: Elements of Generalized Information Theory*, 2nd ed., J. Kacprzyk, Ed. Physica-Verlag, 1999.

[48] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, no. 1, pp. 28–44, 1973.

[49] P. J. King and E. H. Mamdani, "The application of fuzzy control systems to industrial processes," *Automatica*, vol. 13, no. 3, pp. 235–242, 1977.

[50] T. J. Ross, *Fuzzy Logic with Engineering Applications*. John Wiley & Sons, 2004.

[51] H. Hagras, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 524–539, 2004.

[52] J. Mendel, "Type-2 fuzzy sets and systems: an overview," *IEEE Comput. Intell. Mag.*, vol. 2, no. 1, pp. 20–29, 2007.

[53] ——, "Computing with words: Zadeh, turing, popper and occam," *IEEE Comput. Intell. Mag.*, vol. 2, no. 4, pp. 10–17, 2007.

[54] D. Wu and J. Mendel, "A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets," *Information Sciences*, vol. 179, no. 8, pp. 1169–1192, 2009.

[55] R. John, "Type 2 fuzzy sets: an appraisal of theory and applications," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 06, pp. 563–576, 1998.

[56] S. Sebastian and R. John, "Multi-fuzzy sets and their correspondence to other sets," *Annals of Fuzzy Mathematics and Informatics*, 2015.

[57] Y. Yang, S. Liu, and R. John, "Uncertainty representation of grey numbers and grey sets," *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1508–1517, 2014.

[58] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, Z. Xu, B. Bedregal, J. Montero, H. Hagras, F. Herrera, and B. De Baets, "A historical account of types of fuzzy sets and their relationships," *IEEE Trans. Fuzzy Syst.*, 2015.

[59] J. Mendel, H. Hagras, W.-W. Tan, W. W. Melek, and H. Ying, *Introduction to Type-2 Fuzzy Logic Control: Theory and Applications*. John Wiley & Sons, 2014.

[60] H.-J. Zimmermann, *Fuzzy set theory—and its applications*. Springer Science & Business Media, 2001.

[61] G. Klir and B. Yuan, *Fuzzy sets and fuzzy logic*. Prentice Hall New Jersey, 1995, vol. 4.

[62] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence*. Prentice Hall, 1997.

[63] G. Klir, *Uncertainty and information: foundations of generalized information theory*. John Wiley & Sons, 2005.

[64] J.-S. R. Jang, "Self-learning fuzzy controllers based on temporal backpropagation," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 714–723, 1992.

[65] L. X. Wang and J. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 807–814, 1992.

[66] ——, "Back-propagation fuzzy system as nonlinear dynamic system identifiers," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Mar 1992, pp. 1409–1418.

[67] D. Dubois and H. Prade, *Fuzzy sets and systems: theory and applications*. Academic press, 1980, vol. 144.

[68] O. Cordón, F. Herrera, and A. Peregrín, "Applicability of the fuzzy operators in the design of fuzzy logic controllers," *Fuzzy sets and systems*, vol. 86, no. 1, pp. 15–41, 1997.

[69] M. Mizumoto and K. Tanaka, "Some properties of fuzzy sets of type 2," *Information and control*, vol. 31, no. 4, pp. 312–340, 1976.

[70] M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*. Pearson Education, 2005.

[71] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," in *Proceedings of the Institution of Electrical Engineers*, vol. 121, no. 12, 1974, pp. 1585–1588.

[72] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, no. 1, pp. 116–132, 1985.

[73] E. Cox, "Fuzzy fundamentals," *IEEE Spectrum*, vol. 29, no. 10, pp. 58–61, 1992.

[74] C. Wagner and H. Hagras, "Uncertainty and type-2 fuzzy sets and systems," in *2010 UK Workshop on Computational Intelligence (UKCI).*, 2010, pp. 1–5.

[75] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994.

[76] G. Mouzouris and J. Mendel, "Non-singleton fuzzy logic systems," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1994, pp. 456–461.

[77] N. Sahab and H. Hagras, "Adaptive non-singleton type-2 fuzzy logic systems: a way forward for handling numerical uncertainties in real world applications," *Int. J. Comput. Commun. Control*, vol. 5, no. 3, pp. 503–529, 2011.

[78] G. Mouzouris and J. Mendel, "Nonsingleton fuzzy logic systems: theory and application," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 1, pp. 56–71, 1997.

[79] D. Sinha and E. R. Dougherty, "Fuzzification of set inclusion: theory and applications," *Fuzzy sets and systems*, vol. 55, no. 1, pp. 15–42, 1993.

[80] K. Nozaki, H. Ishibuchi, and H. Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," *Fuzzy sets and systems*, vol. 86, no. 3, pp. 251–270, 1997.

[81] T.-P. Hong and C.-Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy sets and Systems*, vol. 84, no. 1, pp. 33–47, 1996.

[82] H. Hellendoorn and C. Thomas, "Defuzzification in fuzzy controllers," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 1, no. 2, pp. 109–123, 1993.

[83] W. Van Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," *Fuzzy sets and systems*, vol. 108, no. 2, pp. 159–178, 1999.

[84] Q. Chen and S. Kawase, "On fuzzy-valued fuzzy reasoning," *Fuzzy sets and systems*, vol. 113, no. 2, pp. 237–251, 2000.

[85] C. Gafa and S. Coupland, "A new recursive type-reduction procedure for general type-2 fuzzy sets," in *IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, 2011, pp. 44–49.

[86] O. Linda and M. Manic, "Importance sampling based defuzzification for general type-2 fuzzy sets," in *IEEE International Conference on Fuzzy Systems (FUZZ)*, 2010, pp. 1–7.

[87] C. Wagner, S. Miller, J. M. Garibaldi, D. T. Anderson, and T. C. Havens, "From interval-valued data to general type-2 fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 2, pp. 248–269, 2015.

[88] M. A. Sanchez, O. Castillo, and J. R. Castro, "Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems," *Expert Systems with Applications*, vol. 42, no. 14, pp. 5904–5914, 2015.

[89] P. Melin, C. Gonzalez, J. R. Castro, O. Mendoza, O. Castillo *et al.*, "Edge-detection method for image processing based on generalized type-2 fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 6, pp. 1515–1525, 2014.

[90] N. Karnik and J. Mendel, "Introduction to type-2 fuzzy logic systems," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, vol. 2, 1998, pp. 915–920.

[91] H. Hassani and J. Zarei, "Interval type-2 fuzzy logic controller design for the speed control of dc motors," *Systems Science & Control Engineering: An Open Access Journal*, vol. 3, no. 1, pp. 266–273, 2015.

[92] S. Turk, R. John, and E. Ozcan, "Interval type-2 fuzzy sets in supplier selection," in *14th UK Workshop on Computational Intelligence (UKCI)*, 2014, pp. 1–7.

[93] J. Mendel, R. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 6, pp. 808–821, 2006.

[94] J. Mendel and R. John, "Footprint of uncertainty and its importance to type-2 fuzzy sets," in *Proc. 6th IASTED Intĺ. Conf. on Artificial Intelligence and Soft Computing*, 2002, pp. 587–592.

[95] N. Karnik and J. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 1, pp. 195–220, 2001.

[96] D. Wu and J. Mendel, "Enhanced karnik–mendel algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 923–934, 2009.

[97] S. Greenfield, F. Chiclana, S. Coupland, and R. John, "The collapsing method of defuzzification for discretised interval type-2 fuzzy sets," *Information Sciences*, vol. 179, no. 13, pp. 2055–2069, 2009.

[98] H. Wu and J. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 5, pp. 622–639, 2002.

[99] D. Wu and W. W. Tan, "Computationally efficient type-reduction strategies for a type-2 fuzzy logic controller," in *The 14th IEEE International Conference on Fuzzy Systems FUZZ'05*, 2005, pp. 353–358.

[100] M. Nie and W. Wan Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2008, pp. 1425–1432.

[101] D. Wu, "An overview of alternative type-reduction approaches for reducing the computational cost of interval type-2 fuzzy logic controllers," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2012, pp. 1–8.

[102] J. Mendel, "On answering the question "where do i start in order to solve a new problem involving interval type-2 fuzzy sets?"," *Information Sciences*, vol. 179, no. 19, pp. 3418–3431, 2009.

[103] J. Mendel and D. Wu, *Perceptual computing: aiding people in making subjective judgments*. John Wiley & Sons, 2010, vol. 13.

[104] O. Linda and M. Manic, "Uncertainty-robust design of interval type-2 fuzzy logic controller for delta parallel robot," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 661–670, 2011.

[105] N. Benatar, U. Aickelin, and J. Garibaldi, "A comparison of non-stationary, type-2 and dual surface fuzzy control," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2011, pp. 1193–1200.

[106] J. Mendel and H. Wu, "Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 1, forward problems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 6, pp. 781–792, 2006.

[107] ——, "Type-2 fuzzistics for nonsymmetric interval type-2 fuzzy sets: forward problems," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 5, pp. 916–930, 2007.

[108] C.-H. Lee and H.-Y. Pan, "Performance enhancement for neural fuzzy systems using asymmetric membership functions," *Fuzzy Sets and Systems*, vol. 160, no. 7, pp. 949–971, 2009.

[109] S. Coupland and R. John, "Geometric interval type-2 fuzzy systems." in *EUSFLAT Conf.*, 2005, pp. 449–454.

[110] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Analysis of the noise reduction property of type-2 fuzzy logic systems using a novel type-2 membership function," *IEEE Trans. Syst., Man, Cybern. B*, vol. 41, no. 5, pp. 1395–1406, 2011.

[111] M. Khanesar, M. Teshnehlab, E. Kayacan, and O. Kaynak, "A novel type-2 fuzzy membership function: application to the prediction of noisy data," in *IEEE Int. Conf. on Comput. Intell. for Measurement Systems and Applications*, 2010, pp. 128–133.

[112] T. Kumbasar and H. Hagras, "A self-tuning zslices based general type-2 fuzzy PI controller," *IEEE Trans. Fuzzy Syst.*, vol. PP, no. 99, p. 1, 2014.

[113] C. Wagner and H. Hagras, "A collection operator for type-2 fuzzy logic systems," in *2010 UK Workshop on Computational Intelligence (UKCI)*, Sept 2010, pp. 1–6.

[114] H. Mahmoodian, S. Salem, and K. Shojaei, "Adaptively adjusted footprint of uncertainty in interval type 2 fuzzy controller for cancer drug delivery," *Procedia Computer Science*, vol. 76, pp. 360–367, 2015.

[115] R. John and S. Coupland, "Type-2 fuzzy logic: challenges and misconceptions [discussion forum]," *IEEE Comput. Intell. Mag.*, vol. 7, no. 3, pp. 48–52, 2012.

[116] D. Wu and W. W. Tan, "A type-2 fuzzy logic controller for the liquid-level process," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, vol. 2, 2004, pp. 953–958.

[117] S. Musikasuwan, T. Ozen, and J. Garibaldi, "An investigation into the effect of number of model parameters on performance in type-1 and type-2 fuzzy logic systems," *Proc. 10th Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU 2004)*, pp. 1593–1600, 2004.

[118] R. D. Jones, Y. Lee, C. Barnes, G. Flake, K. Lee, P. Lewis, and S. Qian, "Function approximation and time series prediction with neural networks," in *Int. Joint Conf. on Neural Networks, IJCNN1990*, 1990, pp. 649–665.

[119] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks : prediction and system modelling," Los Alamos National Laboratory, Los Alamos,NM, Tech. Rep. LA-UR-87-2662, 1987.

[120] J. Jang and C. Sun, "Predicting chaotic time series with fuzzy if–then rules," in *Second IEEE Int. Conf. on Fuzzy Systems*, 1993, pp. 1079–1084.

[121] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, 1993.

[122] J.-S. R. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378–406, 1995.

[123] J. Mendel and G. Mouzouris, "Designing fuzzy logic systems," *IEEE Trans. Circuits Syst. II*, vol. 44, no. 11, pp. 885–895, 1997.

[124] D.-W. Kim and G.-T. Park, "Using interval singleton type 2 fuzzy logic system in corrupted time series modelling," in *Knowledge-Based Intelligent Information and Engineering Systems*, 2005, pp. 566–572.

[125] O. Uncu, K. Kilic, and I. B. Turksen, "A new fuzzy inference approach based on mamdani inference using discrete type 2 fuzzy sets," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, 2004, pp. 2272–2277.

[126] K. Huarng and H.-K. Yu, "A type 2 fuzzy time series model for stock index forecasting," *Physica A: Statistical Mechanics and its Applications*, vol. 353, pp. 445–462, 2005.

[127] S. Musikasuwan, T. Ozen, and J. Garibaldi, “An investigation into the effect of number of model parameters on performance in type-1 and type-2 fuzzy logic systems,” *Proc. 10th Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU 2004)*, pp. 1593–1600, 2004.

[128] M. C. Mackey and L. Glass, “Oscillation and chaos in physiological control systems,” *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[129] K. Ikeda, “Multiple–valued stationary state and its instability of the transmitted light by a ring cavity system,” *Optics Communications*, vol. 30, no. 2, pp. 257–261, 1979.

[130] E. N. Lorenz, “Deterministic nonperiodic flow,” *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[131] S. Park and H. Lee-Kwang, “A designing method for type-2 fuzzy logic systems using genetic algorithms,” in *9th IFSA World Congress and 20th NAFIPS International Conference*, 2001, pp. 2567–2572.

[132] A. Gholipour, B. N. Araabi, and C. Lucas, “Predicting chaotic time series using neural and neurofuzzy models: a comparative study,” *Neural Processing Letters*, vol. 24, no. 3, pp. 217–239, 2006.

[133] B. Rezaie, Z. R. Cherati, M. R. J. Motlagh, and M. Farrokhi, “Stabilization of the lorenz chaotic equations by fuzzy controller,” *Trans. Engin. Comput. Technol*, vol. 15, pp. 246–249, 2006.

[134] S. Musikasuwan, T. Ozen, and J. Garibaldi, “An investigation into the effect of number of model parameters on performance in type-1 and type-2 fuzzy logic systems,” *Proc. 10th Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU 2004)*, pp. 1593–1600, 2004.

[135] G. M. Méndez and M. De los Angeles Hernandez, “Hybrid learning for interval type-2 fuzzy logic systems based on orthogonal least-squares and back-propagation methods,” *Information Sciences*, vol. 179, no. 13, pp. 2146–2157, 2009.

[136] M. Sugeno and T. Yasukawa, “A fuzzy-logic-based approach to qualitative modeling,” *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, 1993.

[137] H. Takagi and I. Hayashi, “NN-driven fuzzy reasoning,” *International Journal of Approximate Reasoning*, vol. 5, no. 3, pp. 191–212, 1991.

[138] S. K. Halgamuge and M. Glesner, “Neural networks in designing fuzzy systems for real world applications,” *Fuzzy sets and systems*, vol. 65, no. 1, pp. 1–12, 1994.

[139] B. Carse, T. C. Fogarty, and A. Munro, “Evolving fuzzy rule based controllers using genetic algorithms,” *Fuzzy sets and systems*, vol. 80, no. 3, pp. 273–293, 1996.

[140] C. Wagner and H. Hagras, “Evolving type-2 fuzzy logic controllers for autonomous mobile robots,” in *Analysis and design of intelligent systems using soft computing techniques (Advances in Soft Computing 41)*. Springer, 2007, pp. 16–25.

[141] F. Herrera, M. Lozano, and J. L. Verdegay, “A learning process for fuzzy control rules using genetic algorithms,” *Fuzzy sets and systems*, vol. 100, no. 1, pp. 143–158, 1998.

[142] D. Wu and W. W. Tan, “Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers,” *Engineering Applications of Artificial Intelligence*, vol. 19, no. 8, pp. 829–841, 2006.

[143] R. Alcalá, J. Casillas, O. Cordón, F. Herrera, and S. J. I. Zwir, “Techniques for learning and tuning fuzzy rule-based systems for linguistic modeling and their application,” in *In C. Leondes (Ed.), Knowledge*, 1999, pp. 889–941.

[144] J. Mendel, “Computing derivatives in interval type-2 fuzzy logic systems,” *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 1, pp. 84–98, 2004.

[145] C.-H. Lee, J.-L. Hong, Y.-C. Lin, and W.-Y. Lai, “Type-2 fuzzy neural network systems and learning,” *International Journal of Computational Cognition*, vol. 1, no. 4, pp. 79–90, 2003.

[146] R. Sepulveda, O. Castillo, P. Melin, O. Montiel, and L. T. Aguilar, “Evolutionary optimization of interval type-2 membership functions using the human evolutionary model,” in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2007, pp. 1–6.

[147] M. Antonelli, P. Ducange, B. Lazzerini, and F. Marcelloni, “Learning concurrently partition granularities and rule bases of mamdani fuzzy systems in a multi-objective evolutionary framework,” *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 1066–1080, 2009.

[148] O. Cordón and F. Herrera, “Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems,” *Fuzzy Sets and Systems*, vol. 118, no. 2, pp. 235–255, 2001.

[149] O. Cordon, F. Herrera, and P. Villar, “Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base,” *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 667–674, Aug 2001.

[150] G. Liu and W. Yang, “Learning and tuning of fuzzy membership functions by simulated annealing algorithm,” in *The 2000 IEEE Asia-Pacific Conference on Circuits and Systems*, 2000, pp. 367–370.

[151] M. Ghazanfari, S. Alizadeh, M. Fathian, and D. Koulouriotis, “Comparing simulated annealing and genetic algorithm in learning fcm,” *Applied Mathematics and Computation*, vol. 192, no. 1, pp. 56–68, 2007.

[152] J. Garibaldi and E. C. Ifeachor, "Application of simulated annealing fuzzy model tuning to umbilical cord acid-base interpretation," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 1, pp. 72–84, 1999.

[153] O. Cordòn, F. Herrera, and P. Villar, "Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing," *International Journal of Approximate Reasoning*, vol. 25, no. 3, pp. 187–215, 2000.

[154] H. Cheng and J.-R. Chen, "Automatically determine the membership function based on the maximum entropy principle," *Information Sciences*, vol. 96, no. 3-4, pp. 163–182, 1997.

[155] G. Liu and W. Yang, "Learning and tuning of fuzzy membership functions by simulated annealing algorithm," in *The 2000 IEEE Asia-Pacific Conference on Circuits and Systems*, 2000, pp. 367–370.

[156] H. Mohamadi, J. Habibi, M. S. Abadeh, and H. Saad, "Data mining with a simulated annealing based fuzzy classification system," *Pattern Recognition*, vol. 41, no. 5, pp. 1824–1833, 2008.

[157] M. Almaraashi and R. John, "Tuning of type-2 fuzzy systems by simulated annealing to predict time series," in *Proceedings of the World Congress on Engineering*, vol. 2, 2011, pp. 976–980.

[158] M. Almaraashi, R. John, S. Coupland, and A. Hopgood, "Time series forecasting using a TSK fuzzy system tuned with simulated annealing," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2010, pp. 1–6.

[159] M. Almaraashi and R. John, "Tuning fuzzy systems by simulated annealing to predict time series with added noise," *Proceedings of UKCI', Essex, UK*, 2010.

[160] M. Almaraashi, R. John, and S. Coupland, "Designing generalised type-2 fuzzy logic systems using interval type-2 fuzzy logic systems and simulated annealing," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2012, pp. 1–8.

[161] M. Almaraashi, R. John, and S. Ahmadi, "Learning of type-2 fuzzy logic systems by simulated annealing with adaptive step size," in *Electrical Engineering and Intelligent Systems*, 2013, pp. 53–64.

[162] M. Almaraashi, R. John, and A. Hopgood, "Automatic learning of general type-2 fuzzy logic systems using simulated annealing," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2014, pp. 2384–2390.

[163] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[164] V. Černỳ, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[165] E. H. L. Aarts and P. J. M. van Laarhoven, "Statistical cooling: A general approach to combinatorial optimization problems," *Philips Journal of Research*, vol. 40, pp. 193–226, 1985.

[166] B. Hajek and G. Sasak, "Simulated annealing — to cool or not," *Systems & Control Letters*, vol. 12, no. 5, pp. 443–447, 1989.

[167] D. F. Wong and C. L. Liu, "A new algorithm for floorplan design," in *Proceedings of the 23rd ACM/IEEE Design Automation Conference*, 1986, pp. 101–107.

[168] C. R. Reeves, Ed., *Modern Heuristic Techniques for Combinatorial Problems*. New York, NY, USA: John Wiley & Sons, Inc., 1993.

[169] D. A. Quinney, *An Introduction to the Numerical Solution of Differential Equations*. New York, NY, USA: John Wiley & Sons, Inc., 1987.

[170] L. X. Wang, *Adaptive fuzzy systems and control: design and stability analysis*. Prentice-Hall, Inc., 1994.

[171] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural computation*, vol. 1, no. 2, pp. 281–294, 1989.

[172] V. Uslan, H. Seker, and R. John, "A support vector-based interval type-2 fuzzy system," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2014, pp. 2396–2401.

[173] T. A. Yanar and Z. Akyürek, "Fuzzy model tuning using simulated annealing," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8159–8169, 2011.

[174] H. Youssef, S. M. Sait, and H. Adiche, "Evolutionary algorithms, simulated annealing and tabu search: a comparative study," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 2, pp. 167–181, 2001.

[175] J. Casillas, O. Cordón, and F. Herrera, "Improving the wang and mendel's fuzzy rule learning method by inducing cooperation among rules," in *Proceedings of the 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference*, 2000, pp. 1682–1688.

[176] L. X. Wang, "The wm method completed: a flexible fuzzy system approach to data mining," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 768–782, 2003.