Jerry M. Mendel

Uncertain Rule-Based Fuzzy Systems

Introduction and New Directions

Second Edition

EXTRAS ONLINE

Springer

Uncertain Rule-Based Fuzzy Systems

Jerry M. Mendel

Uncertain Rule-Based Fuzzy Systems

Introduction and New Directions

Second Edition

Springer

Jerry M. Mendel
University of Southern California
Los Angeles, CA
USA

Additional material to this book can be downloaded from http://www.springer.com/us/book/9783319513690

This book was previously published by: Pearson Education, Inc.

ISBN 978-3-319-51369-0 ISBN 978-3-319-51370-6 (eBook)
DOI 10.1007/978-3-319-51370-6

Library of Congress Control Number: 2017931578

Original book published by Pearson: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions 1st Edition, 2001, 978-0130409690. Rights revision letter attached.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Lotfi A. Zadeh
Father of Fuzzy Sets and Fuzzy Logic

First in scholarship[1],
First in kindness, and
First in the hearts of his colleagues

[1] Paraphrased from General Henry Lee, who said of George Washington: "First in war, first in peace and first in the hearts of his countrymen."

The original version of the book was revised:
For detailed information please see Erratum.
The erratum to the book is available at
https://doi.org/10.1007/978-3-319-51370-6_12

Preface

Uncertainty[2] is the fabric that makes life interesting. For millennia human beings have developed strategies to cope with a plethora of uncertainties, never absolutely sure what the consequences would be, but hopeful that the deleterious effects of those uncertainties could be minimized. This book presents a complete methodology for accomplishing this within the framework of fuzzy sets and systems. This is not the original fuzzy sets and systems, but is an expanded and richer fuzzy sets and systems, one that contains the original fuzzy sets and systems within it.

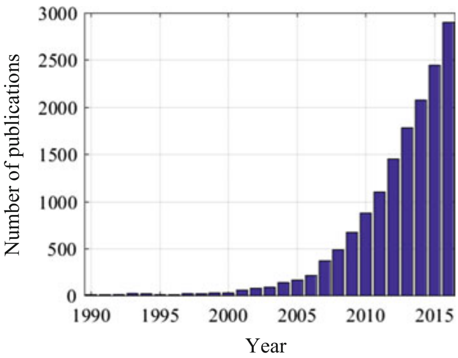
The original fuzzy sets, founded by Prof. Lotfi Zadeh, have been around for more than 50 years, as of the year 2017, and yet the fuzzy systems that use them are unable to handle uncertainties. By *handle*, I mean *to model and minimize the effect of*. That the original fuzzy sets—type-1 fuzzy sets—and the fuzzy systems that use them—type-1 fuzzy systems—cannot do this sounds paradoxical because the word *fuzzy* has the connotation of uncertainty. The expanded fuzzy sets—type-2 fuzzy sets—and the fuzzy systems that use them—type-2 fuzzy systems—are able to handle uncertainties because they can model them and minimize their effects. And, if all uncertainties disappear, type-2 fuzzy sets and systems reduce to their type-1 counterparts, in much the same way that, if randomness disappears, probability and the systems that use it reduce to determinism and deterministic systems.

Although many applications have been found for type-1 fuzzy sets and systems, it is arguably their application to *rule-based systems* that has most significantly demonstrated their importance as a powerful design methodology. Such rule-based systems, both type-1 and type-2, are what this book is about. It explains and shows how to use fuzzy sets and systems in new ways and how to effectively solve problems that are awash in uncertainties.

When the first edition of this book was published in 2001 (Mendel 2001), most of its contents about type-2 fuzzy sets and systems were based on the works of my former Ph.D. students, Nilesh Karnik and Qilian Liang, and mine. Since 2001,

[2]Some of this Preface is taken from or paraphrased from the Preface to the first edition of this book (Mendel 2001).

Fig. 1 The number of publications per year, when searched in Google Scholar using the exact phrase “type-2 fuzzy” excluding citations and patents.[3] The *last bar* is for 2016, up to December 31



thousands of journal and conference articles (Wu and Mendel 2014), as well as some other books, have been published about type-2 fuzzy sets and systems (Fig. 1). Consequently, this new edition is more diverse and presents results developed by many people. As a result of such a large literature, it took me more than seven months just to assemble the materials for this new edition of the book.

One measure of the importance of a field is the number of its publications as well as the citations to them. In addition to these measures, another measure is the number of journal and conference Outstanding Paper Awards that have been given to such papers. The *IEEE Trans. on Fuzzy Systems* has made five such awards to papers that have “Type-2” in their titles (Karnik et al. 1999; Hagras 2004; Coupland and John 2007; Wagner and Hagras 2010; Wu and Mendel 2011). In addition, at least eight other awards have been given to conference or workshop papers about type-2 fuzzy sets and systems between 2005 and 2016.

Fuzzy sets and systems have already been applied in numerous fields, in many of which uncertainties are present (e.g., control, signal processing, digital communications, computer and communication networks, diagnostic medicine, operations research, financial investing, etc.). Hence, the results in this book can immediately be used in all of these fields. To demonstrate the performance advantages for type-2 systems over their type-1 counterparts, when uncertainties are present, I describe and provide results for the following applications in this book: forecasting of time series, knowledge mining using surveys, control, classification of video data working directly with compressed data, and equalization of time-varying nonlinear digital communication channels.

The following major changes have been made from the first to this second edition:

- Mamdani and TSK fuzzy systems are unified.
- Singleton and non-singleton fuzzy systems are also unified.

[3]This figure was prepared by Dongrui Wu, and does not count the number of publications about interval-valued sets and systems.

- α-cuts and their related topics are included because of their importance to general type-2 fuzzy sets and systems.
- Notations about type-2 fuzzy sets are cleaned up.
- Four different and valuable mathematical representations of a general type-2 fuzzy set are explained and used.
- General type-2 fuzzy sets and systems are given much greater prominence.
- A unified approach to type-reduction is presented, one that builds upon a weighted average called the interval-weighted average.
- Mathematical explanations about many aspects of the optimization problems that are associated with type-reduction are included.
- Practical alternatives to type-reduction + defuzzification, called *direct defuzzification*, are presented.
- A case study on fuzzy logic control is included.
- Tables that let the reader know what choices have to be made for the designs of type-1, interval type-2 and general type-2 fuzzy systems are included and illustrated.
- Comprehensive numerical examples are included for type-1, interval type-2 and general type-2 fuzzy systems that illustrate all of their computations.
- Materials that appeared in appendixes at the rear of the first edition are now attached to their respective chapters.
- The appendix in the first edition about Computation has been removed. In its place, Sect. 1.9 refers the reader to sources for downloadable software that will let them implement much of what is in this book.
- Richer and more diverse exercises are included at the end of each chapter.

This book can be read by someone who has an undergraduate BS degree, and should be of great interest to computer scientists and engineers who already use or want to use rule-based systems and are concerned with how to handle uncertainties about such systems. Many worked-out examples are included in the text, and homework exercises are included at the end of Chaps. 2–11 so that the book can be used in a classroom setting as well as a technical reference.

This book can be used for either a one-semester course or a two-semester course. For a *one-semester course*, I would cover:

- Chapter 1: Sects. 1.1 and 1.2
- Chapter 2: All of it except for Sects. 2.11–2.13
- Chapter 3: All of it
- Chapter 6: All of it except for Sects. 6.7.3, 6.7.4 and 6.9
- Chapter 7: Focus only on interval type-2 fuzzy sets and systems. See Sect. 7.1 for a guide on how to do this.
- Chapter 8: Sects. 8.1–8.3.
- Chapter 9: Focus on singleton fuzzification for one kind of interval type-2 fuzzy system, e.g., COS type-reduction + defuzzification for an IT2 Mamdani fuzzy system, Sects. 9.4.1, 9.4.2.1, 9.4.2.4, 9.5, 9.6.3, 9.7, 9.9–9.13
- Design: Sects. 4.1, some of 4.2, 10.1 and some of 10.2

- Application: Choose one of the case studies (forecasting of time series, knowledge mining using surveys or fuzzy logic control)

For a two-semester course, I would cover:

- Semester 1: Chaps. 1–6
- Semester 2: Chaps. 7–11

Portions of this book are an amalgamation of the research of some of my past Ph. D. students who have worked with me during the past 25 years on fuzzy sets and systems. I, therefore, want to give each of them the credit here that they so richly deserve.

Li-Xin Wang studied singleton type-1 fuzzy sets and systems.[4] He developed many concepts about them including the WM method for extracting rules from data, fuzzy basis functions and expansions, tuning the membership function parameters using training data, interpreting a fuzzy system as a layered architecture, and arguably was the first to prove that a certain kind of type-1 fuzzy system is a universal approximator. Many of the topics that are covered in Chap. 3 are due to him.

George Mouzouris extended Li-Xin's works to non-singleton fuzzification, which represented our first attempt at handling one kind of uncertainty (uncertain measurements of the inputs to a fuzzy system) totally within the framework of a fuzzy system. The topics in Chap. 3 about non-singleton fuzzification and fuzzy systems are due to him. He also showed how rule reduction can be achieved by using the SVD algorithm. The material that is in Sect. 4.2.4 is due to him.

Nilesh Karnik provided the entire foundation and framework for singleton type-2 fuzzy systems, including[5] type-reduction and two very widely used algorithms for computing the type-reduced set (the KM algorithms), as well as algorithms for computing the join and meet of general type-2 fuzzy sets, and the extended sup-star composition. Many of the topics that are in Chaps. 6–8 and Sect. 9.4.1 are due to him.

Qilian Liang made type-2 fuzzy systems practical by focusing on how to design such systems when the uncertainties about type-1 fuzzy sets are modeled as type-1 interval fuzzy numbers, the results being interval type-2 Mamdani and TSK fuzzy systems [this was done for singleton and two kinds of non-singleton fuzzification (type-1 and interval type-2)]. Many of the topics that are in Chap. 9 are due to him. He also showed how rule reduction can be achieved in an interval type-2 fuzzy system by using the SVD algorithm (covered in Sect. 10.2). The simulations in this book about time series forecasting of the Mackey–Glass chaotic time series

---

[4]Why I now prefer to call such systems "fuzzy systems" rather than "fuzzy logic systems," as was done in Mendel (2001), is explained in Sect. 1.2.

[5]Although Lotfi Zadeh introduced the concept of a type-2 fuzzy set in 1975, and after that date a very small number of other papers were published about type-2 fuzzy sets, no one prior to our work had developed a type-2 fuzzy system.

(Sects. 4.3 and 10.3), forecasting of compressed video traffic using Mamdani and TSK fuzzy systems (Sects. 4.5 and 10.5), rule-based classification of video traffic (Sects. 4.6 and 10.6), and equalization of time-varying nonlinear digital communication channels (Sect. 10.7), all appeared in the first edition of this book, and were performed by him. I wish to express my sincere appreciation to him for helping me in this way.

Hongwei Wu developed the uncertainty bounds for type-reduced sets. The material that is in Sect. 9.8 is due to her. She also proved a very important result about the switch points of the centroid (covered in Property 8.13).

Feilong Liu developed the α–plane representation of a general type-2 fuzzy set (covered in Sect. 6.7.3) and showed how it can be used to compute the centroid of a general type-2 fuzzy set (covered in Sect. 8.4.1). He also established and proved the properties of the interval-weighted average that are covered in Appendix B.1 in Chap. 8.

Dongrui Wu improved the KM algorithms (the EKM algorithms) and provided many very important insights and theoretical results about fuzzy systems, including continuity of type-1 and interval type-2 fuzzy systems, and fundamental differences between type-1 and interval type-2 fuzzy systems. The topics that are in Sects. 3.9.3, 8.2.4, 9.7, 9.13.2 and 9.13.4 are due to him.

Mohammad Biglarbegian, along with Prof. William Melek, developed the direct defuzzification method known as the BMM method and showed how it can be used in rigorous studies of stability and robustness of a control system. This method is covered in Sect. 9.9.2.

Daoyuan Zhai examined many aspects of general type-2 fuzzy sets, including the connections between endpoints and average of endpoints defuzzification in a general type-2 fuzzy system (covered in Theorem 11.3) and centroid (and enhanced centroid) flow algorithms for speeding up the computations of the centroid of a general type-2 fuzzy set (discussed in Sect. 8.4.1).

Other students who worked on many aspects of type-1 or interval type-2 fuzzy sets and systems that fell outside the scope of this book but that influenced the writing of this book, are Minshen Hao, Mohammad Reza Rajati and Mohammad Mehdi Korjani.

Professor Tufan Kumbasar made very valuable contributions to this book about fuzzy logic control, in Sects. 4.7, 10.8 and 11.16. He wrote much of what is in those three sections, and generated all the simulations. I wish to express my sincere appreciation to him for helping me in this extraordinary way.

I have also had the privilege of working with the Prof. Robert John (the wavy-slice representation of a general type-2 fuzzy set, that is covered in Sect. 6.7.2, is a result of our joint collaboration), Prof. Xinwang Liu (the continuous KM algorithms for the centroid of an interval type-2 fuzzy set and its properties, that are covered in Appendix B.2 of Chap. 8, are results of our joint collaboration), and Prof. Qi-Ye Zhang (who translated the first edition of this book into Chinese in 2014).

During the writing of this second edition, I interacted with the following people who helped me with some technical issues: Prof. Peter Sussner (who was an enormous help in straightening out the notations for type-2 fuzzy sets), Prof. Dongrui Wu, Feilong Liu, Prof. Qilian Liang, Nilesh Karnik, Prof. Tufan Kumbasar and Ondrej Linda. Professor Frank Rhee provided me with Fig. 6.4. All the inputs from these people helped me in reaching the final version of the book. Any remaining errors in content or publishing are my responsibility.

I would also like to acknowledge Prof. Hani Hagras for his many works on extremely interesting and novel applications of interval and general type-2 fuzzy systems, applications that have greatly influenced many other researchers to work on type-2 fuzzy sets and systems.

I gratefully acknowledge material quoted from books or journals by AIAA, American Association for the Advancement of Science, Elsevier, IEEE, McGraw-Hill, Prentice-Hall, Springer, and Wiley. When a quote is used, a reference is made to a specific publication that is listed at the end of each chapter. I also gratefully acknowledge Prof. Vladik Kreinovich for letting me quote some material about similarity and subsethood in Exercises 2.44 and 2.45 from a conference paper (see Nguyen and Kreinovich 2008) that is listed at the end of Chap. 2).

I am also very grateful to my Springer editors Mary James and Zoe Kennedy who guided me through the final production of the book; and to other staff members at Springer for their help in the production of this book.

I want to thank my wife Letty for providing me, for more than 56 years, with a wonderful environment that has made the writing of this book possible. Finally, I want to thank Prof. Lotfi Zadeh, the father of fuzzy logic, for his seminal works without which there would not have been even a first edition of this book and to whom this second edition is dedicated.

Los Angeles, CA, USA — Jerry M. Mendel
January 2017

References

Coupland, S., and R. John. 2007. Geometric type-1 and type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 15 (1): 3–15.

Hagras, H. 2004. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on Fuzzy Systems* 12 (4): 524–539.

Karnik, N.N., J.M. Mendel, and Q. Liang. 1999. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 7 (6): 643–658.

Mendel, J.M. 2001. *Introduction to Rule-Based Fuzzy Logic Systems*. Upper Saddle River, NJ: Prentice-Hall.

Wagner, C., and H. Hagras. 2010. Toward general type-2 fuzzy logic systems based on zslices. *IEEE Transactions on Fuzzy Systems* 18 (4): 637–660.

Wu, D., and J.M. Mendel. 2014. Designing practical interval type-2 fuzzy logic systems made simple. In *Proceedings of 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 800–807, Beijing, China, July 2014.

Contents

Chapter 1
Introduction

1.1 What This Book Is About

This book is about *rule-based systems* that can be used to solve a broad range of problems, from forecasting, to classification, to diagnosis, to judgment making, to control, etc.

A rule has the structure "IF p THEN q," in which p is called the rule's *antecedent* and q is called the rule's *consequent*. A rule-based system begins with a collection of such rules, either extracted from domain-specific data or provided by one or more domain experts. Rule antecedents are in terms of variables that can be observed or measured, and are denoted as $x_i (i = 1, \ldots, p)$. Each rule tells us something about a desired output, denoted by y.

To keep things as simple as possible in this discussion, here are three expert-based rules for when to adjust an air conditioning (AC) unit, in which there is only one antecedent variable, x = temperature, and one output, $y = AC$ adjustment:

$$\begin{cases} \text{IF } x \text{ is moderate, THEN } y = \text{adjust the AC to around low} \\ \text{IF } x \text{ is high, THEN } y = \text{adjust the AC to moderate to high.} \\ \text{IF } x \text{ is very high, THEN } y = \text{adjust the AC to around high} \end{cases} \tag{1.1}$$

A rule-based AC adjustment system, which would be implemented in software or hardware, is activated by measured values of temperature, and automatically adjusts the AC level.

The first thing to notice from the rules in (1.1) is that they use linguistic terms (words) for both temperature and the AC adjustment. Since software and hardware deal with numbers and not words there seems to be a big mismatch between the statement of the rules in (1.1) and being able to implement them so that a measured value of temperature leads to an AC adjustment. Mathematics is needed to quantify these rules and to process a measured value of temperature into a numerical AC adjustment. This book provides this mathematics.

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_1

The second thing to notice from the rules in (1.1) is that the words used in the rules are not very precise, e.g., what does "moderate temperature" mean, or "around low"? Actually, these terms can mean different things to different people and so linguistic uncertainties are present in these rules. Mathematical models are needed that capture linguistic uncertainties. This book provides such mathematical models.

The third thing to notice from these rules is that measurements of temperature (that will activate these rules) may be inaccurate and so mathematical models are needed that account for such inaccuracies. This book also provides the mathematical models to do this.

The fourth thing to notice from these rules is that the words *moderate*, *high*, and *very high* partition temperature into three regions, and so regardless of the nature of rule's consequents (which in other applications could be numbers, functions, categories, words or a mixture of these) the rules that are considered in this book begin by partitioning the inputs over their application-dependent domains. Each consequent fills a rule's specific partition with a number, function, category, or word.

Partitions come in different guises. Four different kinds of partitions of temperature are depicted in Fig. 1.1. The horizontal axis of each figure is a variable such as temperature, and the vertical axis of this figure is the degree of belonging (which is scaled between zero and unity) of each x in a partition, that is also called the *membership* of x in a partition.

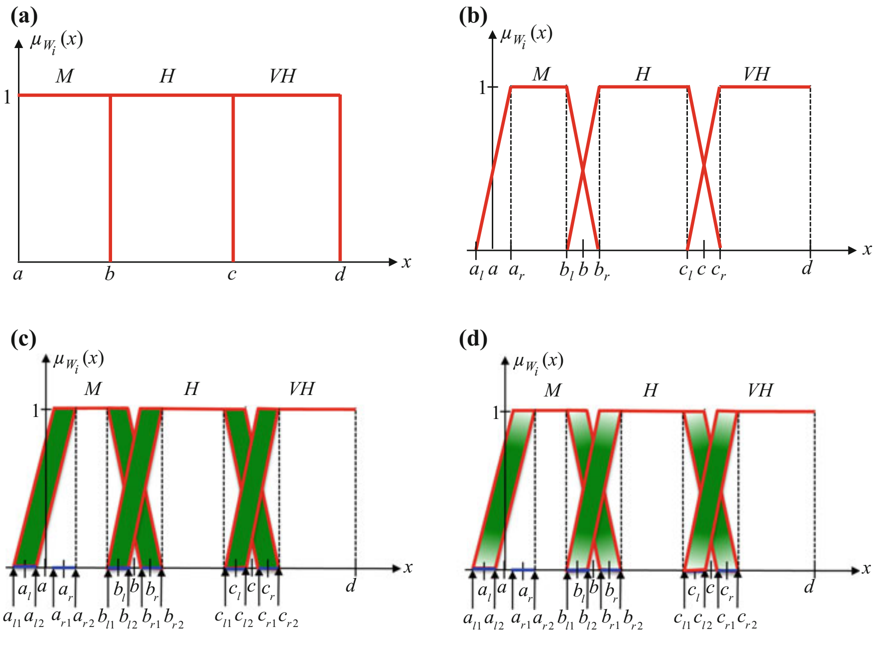


Fig. 1.1 Four kinds of uncertainty partitions: **a** crisp, **b** first-order uncertainty, **c** second-order uncertainty with uniform weighting, and **d** second-order uncertainty with nonuniform weighting. W_i denotes the *i*th word, where $W_1 = M$, $W_2 = H$ and $W_3 = VH$

Suppose, for example, the domain of x is partitioned, as shown in Fig. 1.1a, in agreement with (1.1) into three regions [*moderate* (*M*), *high* (*H*), and *very high* (*VH*)], where the dividing point between each partition is known exactly, in which case no uncertainty exists about $x = a, b, c$ or d. Each of the intervals $[a, b], (b, c], (c, d]$ is a *crisp partition*, and a given value of x can only reside in one of them with full membership in it [which is why the degree of membership, $\mu_{W_i}(x)$, is always unity]. Additionally, each crisp partition is associated with a linguistic term, M or H or VH, and there is always a sharp transition from one term to the next at $x = b$ or c.

Definition 1.1 A *crisp partition* (zero-order uncertainty partition) of the real variable, x, partitions it into nonoverlapping adjacent regions that are intervals of real numbers, where the degree of membership in each region is 1.

Crisp partitions serve us well in some situations, but they do not permit any uncertainty about $x = a, b, c$ or d.

Suppose next that one wants a model that permits uncertainty about $x = a, b$ or c (for the purposes of this discussion, there is no uncertainty about d, e.g., for the AC example, $d \equiv 48\,°\mathrm{C}$). This uncertainty can be expressed by letting each number about which there is uncertainty become an interval of numbers, i.e., $a \to [a_l, a_r]$, $b \to [b_l, b_r]$ and $c \to [c_l, c_r]$. Figure 1.1a now changes to Fig. 1.1b. Each of the intervals $[a_l, b_r], [b_l, c_r]$and$[c_l, d]$ is associated with the regions M, H, and VH, respectively, and can be called a *first-order uncertainty partition*. Observe in Fig. 1.1b that:

- For $x \in [a_l, a_r]$, x is in the region called M and its degree of membership in M rises from a value of 0 when $x = a_l$ to a value of 1, when $x = a_r$; it is the non-unity value of the degree of membership of $x \in [a_l, a_r)$ that models the first-order uncertainty about a.
- For $x \in [a_r, b_l]$, x is in the region called M and its degree of membership in M is 1, so there is no uncertainty about x being in region M when $x \in [a_r, b_l]$; by comparing Fig. 1.1a, b, observe that first-order uncertainty about a reduces the length of the interval of no uncertainty about being in region M, i.e., $|b_l - a_r| < |b - a|$.
- For $x \in [b_l, b_r]$, x is simultaneously in the regions called M and H, but to different degrees of membership (except at the single point where the positive-sloping and negative-sloping lines intersect); it is the non-unity value of the degree of membership of $x \in [b_l, b_r)$ that models the uncertainty about point b, but in Fig. 1.1a because point b separates the regions M and H, when $b \to [b_l, b_r]$ both of these regions inherit some of the first-order uncertainty about b.
- For $x \in [b_r, c_l]$, x is in the region called H and its degree of membership in H is 1, so there is no uncertainty about x being in region H when $x \in [b_r, c_l]$; by comparing Fig. 1.1a, b, observe that first-order uncertainties about points b and c reduce the length of the interval of no uncertainty about being in region H, i.e., $|c_l - b_r| < |c - b|$.

- For $x \in [c_l, c_r]$, x is simultaneously in the regions called H and VH, but to different degrees of membership (except at the single point where the positive-sloping and negative-sloping lines intersect); it is the non-unity value of the degree of membership of $x \in [c_l, c_r)$ that models the uncertainty about point c, but in Fig. 1.1a because point c separates the regions H and VH, when $c \rightarrow [c_l, c_r]$ both of these regions inherit some of the first-order uncertainty about c.

Definition 1.2 A *first-order uncertainty partition* of the real variable, x, partitions it into overlapping intervals, where one is absolutely certain about where the overlap begins and ends, so that the degree of membership in each region of overlap is a real number that is an element of $[0, 1]$.

Overlapping endpoint intervals lead to smooth transitions from one region (linguistic term) to another, which is very different from the sharp transitions that occur when crisp partitions are used. First-order uncertainty partitions serve us well in many situations, but they do not allow for any uncertainty about the overlap.

Uncertainty about the interval endpoints of a first-order uncertainty partition is called *second-order uncertainty*; it can once again be expressed by letting each number about which there is uncertainty become an interval of numbers, i.e., $a \rightarrow \left[[a_{l1}, a_{l2}], [a_{r1}, a_{r2}]\right]$, $b \rightarrow \left[[b_{l1}, b_{l2}], [b_{r1}, b_{r2}]\right]$ and $c \rightarrow \left[[c_{l1}, c_{l2}], [c_{r1}, c_{r2}]\right]$. Figure 1.1b now changes to Fig. 1.1c. Each of the intervals $[a_{l1}, b_{r2}], [b_{l1}, c_{r2}]$ and $[c_{l1}, d]$ is associated with the regions M, H, and VH, respectively, and can be called a *second-order uncertainty partition*. Observe in Fig. 1.1c that:

- For $x \in [a_{l1}, a_{r2}]$, x is in the region called M and its degree of membership in M is an interval of values; it is the interval nature of the degree of membership of $x \in (a_{l1}, a_{r2})$ that models the second-order uncertainty about a.
- For $x \in [a_{r2}, b_{l1}]$, x is in the region called M and its degree of membership in M is 1, so there is no uncertainty about x being in region M when $x \in [a_{r2}, b_{l1}]$; by comparing Fig. 1.1b, c, observe that second-order uncertainty about a further reduces the length of the interval of no uncertainty about being in region M, i.e. $|b_{l1} - a_{r2}| < |b_l - a_r|$.
- For $x \in [b_{l1}, b_{r2}]$, x is simultaneously in the regions called M and H, but to different interval-valued degrees of membership; it is the interval nature of the degree of membership of $x \in (b_{l1}, b_{r2})$ that models the uncertainty about point b, but in Fig. 1.1a because point b separates the regions M and H, when $b \rightarrow \left[[b_{l1}, b_{l2}], [b_{r1}, b_{r2}]\right]$ both of these regions inherit some of the second-order uncertainty about b.
- For $x \in [b_{r2}, c_{l1}]$, x is in the region called H and its degree of membership is 1, so there is no uncertainty about x being in region H when $x \in [b_{r2}, c_{l1}]$; by comparing Fig. 1.1b, c, observe that second-order uncertainty about points b and c further reduce the length of the interval of no uncertainty about being in region H, i.e., $|c_{l1} - b_{r2}| < |c_l - b_r|$,

- For $x \in [c_{l1}, c_{r2}]$, x is simultaneously in the regions called H and VH, but to different interval-valued degrees of membership; it is the interval nature of the degree of membership of $x \in (c_{l1}, c_{r2})$ that models the uncertainty about point c, but in Fig. 1.1a because point c separates the regions H and VH, when $c \to [[c_{l1}, c_{l2}], [c_{r1}, c_{r2}]]$ both of these regions inherit some of the second-order uncertainty about c.

Definition 1.3 A *second-order uncertainty partition* of the real variable, x, partitions it into overlapping intervals, where one is unsure about where the overlap begins and ends, so that the degree of membership in each region of overlap is an interval of real numbers that is a subset of $[0, 1]$.

Definition 1.4 Each region in $X \times \{\mu_{W_i}(x)\}$, in which the degree of membership is an interval of real numbers, is called the *footprint of uncertainty* (FOU).

Definition 1.5 A uniformly shaded FOU (Fig. 1.1c) denotes a uniform weighting of all of its points, and is called a *uniformly weighted second-order uncertainty partition*.

Definition 1.6 A nonuniformly shaded FOU (Fig. 1.1d) denotes a nonuniform weighting of all of its points, and is called a *nonuniformly weighted second-order uncertainty partition*.

Readers are no doubt already anticipating additional levels of uncertainty, along the lines just given, e.g., $b \to \Big[\big[[b_{l1_l}, b_{l1_r}], [b_{l2_l}, b_{l2_r}]\big], \big[[b_{r1_l}, b_{r1_r}], [b_{r2_l}, b_{r2_r}]\big]\Big]$. However, because

$$\Big[\big[[b_{l1_l}, b_{l1_r}], [b_{l2_l}, b_{l2_r}]\big], \big[[b_{r1_l}, b_{r1_r}], [b_{r2_l}, b_{r2_r}]\big]\Big] \Leftrightarrow \big[[b_{l1_l}, b_{l2_r}], [b_{r1_l}, b_{r2_r}]\big] \quad (1.2)$$

a second-order uncertainty model suffices.[1]

It is one thing to describe the different kinds of uncertainty models using simple pictures, as has just been done, but it is another thing to do all of this using mathematics, which is what needs to be done in order to implement a rule-based system. The really good news is that one does not need to invent new mathematics to do all of this; it already exists and is *set theory*.

Crisp partitions can be described mathematically using classical (crisp) set theory; first-order uncertainty partitions can be described mathematically using classical (type-1) fuzzy set theory; uniformly weighted second-order uncertainty partitions can be described mathematically using interval type-2 fuzzy set theory; and nonuniformly weighted second-order uncertainty partitions can be described mathematically using general type-2 fuzzy set theory. This book covers all of these

[1]It is conceivable that uncertainty about the filling of the FOU could lead to higher than second-order uncertainty about the FOU.

set-theoretic models, but places almost all of its emphasis on the three kinds of fuzzy set models, because it is only those models that model first- and second-order uncertainty partitions.

This book's approach to covering these set theory models is *bottom-up*. Classical sets will be shown to give rise to type-1 fuzzy sets, which in turn give rise to interval type-2 fuzzy sets, which in turn give rise to general type-2 fuzzy sets. Knowledge learned about, and algorithms developed for a lower level set will be used by a higher level set, and so the time and effort spent learning about a lower level set will be well spent.

Example 1.1 Some people may object in principle to using fuzzy sets instead of the more classical crisp sets. For the purposes of this example, a fuzzy set can be treated as one whose members have a membership grade in it that can be a real number in $[0, 1]$. To dispel the notion of crispness (i.e., dual-valued concepts, which are either true or not true, and that have a membership grade of 1 if true and 0 if not true), a collection of terms is listed in Table 1.1; these terms are widely used in control, signal processing, and communications. While one frequently strives for crisp values of these terms, one usually uses them in contexts where the linguistic terms actually convey more useful information than would a crisp value.

Correlation is an interesting example, because it can be defined mathematically so that, for a given set of data, one can compute a crisp number for it. Let us assume that correlation has been normalized so that it can range between zero and unity, and that for a given set of data one computes the correlation value as 0.15. When explaining the amount of data correlation to someone else, it is usually more meaningful to explain it as "these data have low correlation." Doing this, one is actually fuzzifying the crisp value of 0.15 into the fuzzy set "low correlation."

Stability is another very interesting example. A system is either stable or not stable; there is nothing fuzzy about this. However, if the system is stable, one frequently describes its degree of relative stability, using any of the terms listed in Table 1.1. These terms may be more meaningful than the following description:

Table 1.1 Engineering terms whose contextual usage is usually quite fuzzy

Term	Contextual usage
Alias	None, a bit, high
Bandwidth	Narrowband, broadband
Blur	Somewhat, quite, very
Correlation	Low, medium, high, perfect
Errors	Large, medium, small, a lot of, not so great, very large, very small, almost zero
Frequency	Low, high, ultra-high
Resolution	Low, medium, high
Sampling	Low rate, medium rate, high rate, very high rate
Stability	Stable (lightly damped, highly damped, over damped, critically damped), unstable

[Mendel (1995)] © 1995, IEEE)

The system has four complex poles and the effective damping ratio for the system is 0.3. Just describe the response of such a system as “lightly damped.” Doing this, one is fuzzifying the crisp value of 0.3 into the fuzzy set “lightly damped.”

1.2 The Structure of a Rule-Based Fuzzy System

A *rule-based fuzzy system* contains four components—*rules*, *fuzzifier*, *inference*, and *output processor*—that are interconnected as shown in Fig. 1.2. Once the rules have been established, the fuzzy system can be viewed as a mapping from inputs to outputs (the red path in Fig. 1.2), and this mapping can be expressed quantitatively as $y = f(\mathbf{x})$. This fuzzy system is also known as a fuzzy logic system (Mendel 2001), fuzzy-rule-based system, fuzzy expert system, fuzzy model, or fuzzy logic controller (Jang and Sun 1995), (Jang et al. 1997). In this book, rule-based fuzzy system is shortened to *fuzzy system*.

Rules are the heart of a fuzzy system, and, as already mentioned, they are either extracted from domain-specific data or are provided by one or more domain experts. Fuzzy sets model the linguistic terms that appear in the antecedents or consequents of rules. When type-1 fuzzy sets are used, the fuzzy system is called a *type-1 fuzzy system*; when at least one interval type-2 fuzzy set is used, the fuzzy system is called an *interval type-2 fuzzy system*; and when at least one general type-2 fuzzy set is used, the fuzzy system is called a *general type-2 fuzzy system*. Regardless of what kinds of fuzzy sets are used to model the antecedents or consequents, the rules remain the same. Paraphrasing the American author Gertrude Stein, who wrote “A rose is a rose is a rose… is a rose,” one can say that “A rule is a rule is a rule … is a rule.”

Rules are quantified using the mathematics of fuzzy sets, and that mathematics is different for type-1, interval type-1, and general type-2 fuzzy sets. The quantified rules do nothing until they are activated by measured values of their antecedent variables (in much the same way that an automobile does nothing until its engine is turned on and gasoline is injected into it). It is this activation that leads to the outputs of the fuzzy system.

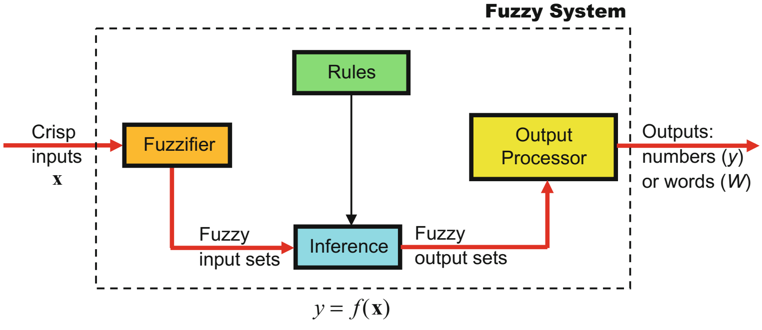


Fig. 1.2 Fuzzy system

The activation of the fuzzy system begins with measured values of the inputs $\{x_i\}_{i=1}^p \equiv \mathbf{x}$. These measured values are real numbers,[2] and to make them mathematically commensurate with the antecedents of the rules, which have been modeled as fuzzy sets, each x_i has to be converted into a fuzzy set (so that one can then go to the rule's consequent). This is done by the *fuzzifier* block in Fig. 1.2, and is called *fuzzification*. For a type-1 fuzzy system, the fuzzifier maps each x_i into a type-1 fuzzy set; for an interval type-2 fuzzy system, the fuzzifier maps each x_i into an interval type-2 fuzzy set; and, for a general type-2 fuzzy system, the fuzzifier maps each x_i into a general type-2 fuzzy set. Fuzzification also depends on whether or not x_i is measured perfectly or if the measurements are corrupted by noise, and in the latter case if the noise is stationary or nonstationary.

The *inference* block (also called an *inference engine*) maps fuzzy sets into fuzzy sets,[3] and this mapping is different for type-1, interval type-2, or general type-2 fuzzy systems. The original fuzzy logic mechanisms for inference were motivated by the inference mechanisms of classical (crisp) logic; however, because the latter mechanisms had problems when they were converted to fuzzy logic inference mechanisms (this is explained in Sect. 2.17), they were modified to make them more practical for real-world applications.[4] Because the modified inference mechanisms no longer have a strong direct connection to logic, the phrase "fuzzy logic system," that was used throughout the first edition of this book (Mendel 2001), has been shortened in this book to "fuzzy system."

More than one rule can be fired by the inference engine due to overlapping fuzzy sets (Fig. 1.1b–d), so a decision must be made about what to do when this occurs. One approach is to combine the outputs of the fired rules, whereas another approach is not to do this. In this book, it is assumed that the choice about what to do when more than one rule is fired is made ahead of time by the end user as part of the design of a fuzzy system, and that this decision is then incorporated into the inference engine.

In many real-world applications of a fuzzy system, crisp numbers must be obtained at its output, e.g., in a control system application such a number corresponds to a control action, in a data processing application such a number corresponds to the prediction of next year's sun spot activity, or to a financial forecast, or to the location of a target, or to the classification of an individual as a terrorist, etc. Obtaining a crisp number at the output of a fuzzy system is accomplished in the *output processor* block. For a type-1 fuzzy system, output processing is done in one stage called *defuzzification*, and is a mapping from a type-1 fuzzy set into a number. For a type-2 fuzzy system, there can be two different kinds of output processors:

---

[2][Dick (2005)] and [Chen et al. (2010)] develop fuzzy systems for complex numbers, but such systems are beyond the scope of this book.

[3]Stating that the outputs of the inference engine are fuzzy sets is very general, and is meant to include everything from numbers, to intervals, to type-1 fuzzy sets, to interval type-2 fuzzy sets, and to general type-2 fuzzy sets. This will be clarified in Chaps. 3, 9, and 11.

[4]The unmodified fuzzy logic inference mechanisms are still being used, e.g., in approximate reasoning applications, but their use is outside of the scope of this book.

(1) A two-stage processor in which type-2 fuzzy sets are first converted into type-1 fuzzy sets, by a process called *type-reduction*, after which the resulting type-1 fuzzy sets are defuzzified; and (2) A one-stage processor called *direct defuzzification*, in which type-2 fuzzy sets are directly defuzzified into a number.

Figure 1.2 shows that the output of a fuzzy system can also be a word. Such an output is popular when a rule-based system is used in real-world applications having to do with subjective judgment making (e.g., what level of pollution is occurring in an environmental situation). For such applications, the output processor has to map the fuzzy set or sets that are at the output of the inference block back into a linguistic term. This kind of output processing is sometimes called *decoding* [e.g., Mendel and Wu (2010)].

1.3 A New Direction for Fuzzy Systems

Type-2 fuzzy systems move the world of fuzzy systems into a fundamentally new and important direction. What is this new direction and why is it important? To make the answers to these questions as clear as possible, consider the following brief digression that reviews some things that are, no doubt, familiar to the reader.

Probability theory is used to model random uncertainty, and within that theory one begins with a probability density function (pdf) that embodies total information about random uncertainties. In most practical real-world applications it is impossible to know or determine the pdf, so one falls back on using the fact that a pdf is completely characterized by all of its moments (if they exist). If the pdf is Gaussian, then, as is well known, two moments—the mean and variance—suffice to completely specify it. For most pdfs, an infinite number of moments are required. Of course, it is not possible, in practice, to determine an infinite number of moments; so, instead, one computes as many moments as are believed to be necessary to extract as much information as possible from the data. At the very least, one uses two moments, the mean and variance, and, in some cases, even higher-than-second-order moments are used.

To use just the first-order moments would not be very useful, because random uncertainty requires an understanding of (at the very least) dispersion about the mean, and this information is provided by the variance. So, the accepted probabilistic modeling of random uncertainty focuses to a large extent on methods that use at least the first two moments of a pdf. This is, for example, why designs based on minimizing a mean-squared error are so popular.

Should one expect any less of a fuzzy system for linguistic uncertainties or any other types of uncertainties? To date, the output of a type-1 fuzzy system may be viewed as analogous to the mean of a pdf. Just as variance provides a measure of dispersion about the mean, and is almost always used to capture more about probabilistic uncertainty in practical statistical-based designs, a fuzzy system also needs some measure of dispersion—*the new direction*—to capture more about its uncertainties than just a single number. Type-2 fuzzy sets provide this measure of dispersion and (I hope to convince you) seem to be as fundamental to the design of systems which include linguistic and/or numerical uncertainties that translate into rule or input uncertainties, as variance is to the mean—*the importance of the new direction*.

1.4 Fundamental Design Requirement

Behind everything that is done in this book is the following *fundamental design requirement* [Karnik and Mendel (1998a, b)]:

> When all sources of membership function uncertainty disappear, a type-2 fuzzy set must reduce to a type-1 fuzzy set, and a type-2 fuzzy system must reduce to a comparable type-1 fuzzy system.

So, for example, when all uncertainty disappears, the extended sup-star composition (Chap. 7) reduces to the usual sup-star composition (Chap. 2) and type-reduction (Chap. 8) reduces to defuzzification (Chap. 3). In this way, a type-2 fuzzy system represents a generalization of a type-1 fuzzy system and not a replacement.

This design requirement is analogous to what happens to a probability density function when random uncertainties disappear. In that case, the variance of the pdf goes to zero, and a probability analysis reduces to a deterministic analysis. So, just as the capability for a deterministic analysis is embedded within a probability analysis, the capability for a type-1 fuzzy system is embedded within a type-2 fuzzy system.

1.5 An Impressionistic[5] Brief History of Type-1 Fuzzy Sets and Fuzzy Logic

Professor Lotfi Zadeh is the founding father of fuzzy sets and fuzzy logic (a short biography of him appears in Sect. 2.2.1). His first seminal paper on fuzzy sets appeared in 1965 (Zadeh 1965), although he began to formulate ideas about them at least four years earlier. Fuzzy sets met with great resistance in the West, perhaps because of the negative connotations associated with the word "fuzzy." Let's face it, the word "fuzzy" does not conjure up visions of scientific or mathematical rigor.

After 1965 some people, along with Zadeh, developed the rigorous mathematical foundations of type-1 fuzzy sets and fuzzy logic. Interestingly enough, Chinese and Japanese researchers devoted a large effort to fuzzy sets and fuzzy logic. A popular hypothesis for this is that "fuzzy" fits in quite nicely with Eastern philosophies and

---

[5]In literature, *impressionism* is a "mode of treatment in which scene, character, and emotion are depicted through the author's or character's impressions rather than by strict objective detail." [*New Webster's Dictionary of the English Language*, Delair Publ. Co., 1981].

religions (e.g., the complementarity of Yin and Yang). But, until the early 1970s fuzzy logic was a theory looking for an application. Then, a major breakthrough occurred in 1974 and 1975, when Mamdani and Assilian (Mamdani 1974; Mamdani and Assilian 1975) showed how to use a rule-based fuzzy system to control a nonlinear dynamical system. It was relatively easy to do this, used rules extracted from experts, and was a fast way to design a control system. Although the design did not lend itself to the well-accepted, important, critical, and rigorous examinations called for by control theory, it did demonstrate an important real application for a rule-based fuzzy system.

Other applications of rule-based fuzzy systems began to appear, two very notable ones in Japan—control of the Sendai cities' subway system, and control of a water treatment system. Commercial products began to appear, e.g., fuzzy washing machine, fuzzy rice cooker, fuzzy shower. In Japan, the word "fuzzy" took on the connotation of "intelligent," and in 1990 Zadeh received a major award. Western industries took notice and the decade of the '90s rolled in, during which fuzzy sets and systems achieved a high degree of acceptability.

The journal *Fuzzy Sets and Systems* was launched in 1978. In 1992, the Institute of Electrical and Electronic Engineers (IEEE) launched the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), and in 1993 the IEEE launched the *IEEE Transactions on Fuzzy Systems*. According to an email sent by Prof. Zadeh (dated July 2, 2016), as of 2016 there are 33 journals with fuzzy in their titles. Many other journals also publish articles about fuzzy sets and systems (e.g., *Information Sciences*, *IEEE Trans. on Systems, Man, and Cybernetics*, *Applied Soft Computing*, *International Journal of Approximate Reasoning*, *Granular Computing*, etc.). There also are many annual international workshops and conferences devoted either exclusively to or that include sessions on fuzzy technologies.

In 1995, the IEEE awarded Prof. Zadeh its highest honor, its Medal of Honor, which is comparable to the Nobel Prize.

Fuzzy sets and systems are now widely used in many industries and fields to solve practical problems, and are still subjects of intense research by academics all over the world. Although many applications have been found for fuzzy sets and systems, it is arguably its application to rule-based systems that has most significantly demonstrated its importance as a powerful design methodology. Such rule-based fuzzy systems are what this book is all about.

If you are interested in a less impressionistic history of fuzzy sets and systems, then see, McNeill and Freilberger (1992), Wang (1997) or Yen and Langari (1999, pp. 3–18).

1.6 Literature on Type-2 Fuzzy Sets and Fuzzy Systems

Many books have been published that are devoted exclusively to fuzzy sets and systems or have one or more chapters about them. Since this book is devoted exclusively to them, I want to let you know about the others that are also devoted exclusively to them. Because there are so many, my criterion for providing the following list is that they either have been cited in Google Scholar at least 2000 times or have "type-2" in their title.

- The following six books have been cited in Google Scholar at least 2000 times (as of November 2016), and are in decreasing order of citations: Klir and Yuan (1995), Dubois and Prade (1980), Ross (2004), Wang (1997), Mendel (2001) and Yager and Filev (1994). The first two books have some very modest discussions about type-2 fuzzy sets, the third, fourth, and sixth books have nothing about them, and the fifth book is the first edition of the present book. Except for Mendel (2001) none of the other books has anything about type-2 fuzzy systems.
- The following books have "type-2" in their titles: Aliev and Guirimov (2014), Castillo (2012), Castillo and Melin (2008, 2012), Cervantes and Castillo (2016), Melin (2012), Mendel, et al. (2014) and Sadeghian et al. (2013).

When the first edition of this book was written (Mendel 2001), during the years 1999 and 2000, type-2 fuzzy sets and systems were in their infancies and there were a relatively small number of articles about them (around 1999 there were less than 40 that had anything to do with type-2 fuzzy sets or systems). Now, more than 17 years later there are thousands of articles about them (see Fig.1 in the Preface).[6] Obviously, it is not possible to include all of these references; so, to those authors whose works have not been included herein my sincere apologies. Today, it is very easy to go to the Internet to find all of these articles.

This section briefly reviews the early literature (1975–1992), the literature that was used very heavily when the first edition of this book was written, and some literature about applications of type-2 fuzzy sets and systems. As one progresses through this book, the reader will be directed to the many references from 2001 to 2016 that have influenced its writing.

1.6.1 Early Literature: 1975–1992

Zadeh (1975) introduced the concept of *fuzzy sets of type-2*, later shortened by others to *type-2 fuzzy sets*, as an extension of an ordinary fuzzy set, i.e., a type-1 fuzzy set. Mizumoto and Tanaka (1976) studied the set-theoretic operations of type-2 fuzzy sets and properties of membership grades of such sets; Mizumoto and Tanaka (1981) also examined type-2 fuzzy sets under the operations of algebraic product and algebraic sum. Nieminen (1977) provided more detail about the algebraic structure of type-2 fuzzy sets. Dubois and Prade (1978), (1979), (1980)

[6]An excellent historical view of type-2 fuzzy sets and systems is John and Coupland (2007). It includes a figure with the number of type-2 related publications over time from 1976 through 2006 as well as a figure that depicts a time line of the historical development of type-2 fuzzy sets and systems.

discussed fuzzy-valued logic and gave a formula for the composition of type-2 relations as an extension of the type-1 sup-star composition, but their formula is only for the minimum t-norm. Hisdal (1981) studied rules and interval sets for higher-than-type-1 fuzzy logic.

Zadeh (1975) also introduced the concept of *fuzzy sets with interval value membership functions*, later shortened by others to *interval-valued fuzzy sets* (IVFS). According to Bustince et al. (2016):

> In 1975 Sambuc (1975) presented in his doctoral thesis[7] the concepts of an interval-valued fuzzy set named a ϕ fuzzy set. In the same year, Jahn (1975) wrote about the notion of interval-valued fuzzy set. One year later Grattan-Guinness (1976) established the definition of an interval-valued membership function. In that decade interval-valued fuzzy sets appeared in the literature in various guises and it was not until the 1980, with the work of Gorzalczany (1987), (1988), (1989a, b), Dziech and Gorzalczany (1987), Türksen (1986, 1992), Türksen and Yao (1984)] and Türksen and Zhong (1990) that the importance of these sets, as well as their name was definitely established.

The timing (Mendel 2010) was not so good for these early developers of type-2 fuzzy sets and IVFSs because the world was paying attention mainly to fuzzy logic control using type-1 fuzzy sets, and these more advanced fuzzy sets seemed to be way ahead of their time, since computation for such sets using 1970–1980s computers was slow and quite limited. Consequently, there was a period of time during which not much, if anything, appeared about type-2 fuzzy sets.

Finally, why did it take so long for the concept of a type-2 fuzzy set to emerge? According to Mendel (2007):

> It seems that science moves in progressive ways where one theory is eventually replaced or supplemented by another, and then another. In school we learn about determinism before randomness. Learning about type-1 fuzzy sets before type-2 fuzzy sets fits a similar learning model. So, from this point of view it was very natural for type-1 fuzzy sets to have been developed as far as possible. Only by doing so was it really possible later to see the shortcomings of such fuzzy sets when one tries to use them to model words or to apply them to situations where uncertainties abound. … Recall, also, that in the 1970s people were first learning what to do with type-1 fuzzy sets, e.g. fuzzy logic control. Bypassing those experiences would have been unnatural. Once it was clear what could be done with type-1 fuzzy sets, it was only natural for people to then look at more challenging problems.

1.6.2 Publications that Heavily Influenced the First Edition of This Book

Karnik and Mendel (1998a, b), (2001a) extended the works of Mizumoto and Tanaka to obtain practical algorithms for performing union, intersection, and complement for type-2 fuzzy sets; developed the concepts of type-reduction,

[7]This material about IVFSs was written in French, apparently never published in a refereed journal, and so it was not, and still is not available in English to the general scientific community.

centroid, and generalized centroid of type-2 fuzzy sets; provided two practical algorithms for computing the latter two for interval type-2 fuzzy sets in Karnik and Mendel (1998b, 2001b); obtained a general formula for the extended sup-star composition of type-2 relations in Karnik and Mendel (1998b, c), Karnik and Mendel (1999); and, based on this formula, Karnik and Mendel (1998a, b, c), Karnik and Mendel (1999) established a complete type-2 fuzzy [logic] system theory[8], one that included type-reduction, although the latter was developed first in Karnik and Mendel (1998b, c).

Liang and Mendel (2000a, c) developed a complete theory for Mamdani interval type-2 fuzzy [logic] systems, for different kinds of fuzzifiers, and showed how such fuzzy systems could be designed, i.e., how the free parameters within interval type-2 fuzzy systems could be tuned by using training data. They also developed type-2 TSK fuzzy [logic] systems in Liang and Mendel (1999), (2001).

There were also some articles about type-2 fuzzy sets that appeared in the Japanese literature, but are only in Japanese. Two examples are Izumi et al. (1983) and Sugeno (1983). No doubt, there are other articles that have eluded this author, and, if there are, I hope that we hear about them.

1.6.3 Application Papers

The first edition of this book (Mendel 2001) stated that, as of 2001, type-2 fuzzy sets and systems have been used for:[9]

- Co-channel interference elimination from nonlinear time-varying communication channels in Liang and Mendel (2000f)
- Connection admission control in Liang et al. (2000)
- Control of mobile robots in Wu (1996)
- Decision making in Yager (1980), Chaneau et al. (1987)
- Equalization of nonlinear fading channels in Karnik and Mendel (1999), Mendel (2000), Liang and Mendel (2000b, d)
- Extracting knowledge from questionnaire surveys in Karnik and Mendel (1998b, 1999b), Liang et al. (2000)
- Forecasting of time-series in Karnik and Mendel (1999), Mendel (2000), Liang and Mendel (2000c)
- Function approximation in Karnik and Mendel (1998b)

[8]These are Mamdani type-2 fuzzy systems. The two most popular fuzzy systems used by engineers are the *Mamdani* and *Takagi-Sugeno-Kang* (TSK) systems (see Chap. 3). Both are characterized by IF–THEN rules and have the same antecedent structures. They differ in the structures of the consequents. The consequent of a rule in a Mamdani fuzzy system is a fuzzy set, whereas the consequent of a rule in a TSK fuzzy system is a mathematical function.

[9]This list is in alphabetical order by application.

- Modeling and Classification of coded video streams in Liang and Mendel (2000g, 2001)
- Pre-processing radiographic images in John (1996, 1998)
- Relational databases in Chiang et al. (1997)
- Solving fuzzy relation equations in Wagenknecht and Hartmann (1988)
- Transport scheduling in John (1996)

Since 2001, there have been an enormous number of real-world applications for type-2 fuzzy sets and systems. In chronological order:

- Mendel (2007) provides references for the following applications: approximation, control, databases, decision-making, embedded agents, health care, hidden Markov models, neural networks, noise cancelation, pattern classification, quality control, spatial query, and wireless communication.
- Hagras (2007) provides references for and discussions about applications of interval type-2 fuzzy logic control to industrial control, mobile robot control, and ambient intelligent environments [see, also, Hagras et al. (2015)].
- Dereli et al. (2011) provide references for and discussions about the following applications: manufacturing operations and industries, service operations and industries, and information and communication technology; its tables include why the particular application required type-2 fuzzy sets.
- Hagras and Wagner (2012) provide references for and discussions about a set of sample applications of type-2 fuzzy systems in the following sectors: business and finance, electrical energy, automatic control, networks, and medical/health.
- Castillo and Mellin (2014) provide many references for and discussions about applications of interval type-2 fuzzy systems in intelligent control; its tables also include why type-2 fuzzy sets were required for the specific application.
- Mendel et al. (2014, Sect. 1.8) provide references for and discussions about real-world applications of interval type-2 Mamdani fuzzy logic controllers in: industrial control, airplane altitude control, control of mobile robots, and control of ambient intelligent environments.

1.7 Coverage

Chapter 2 formally introduces type-1 fuzzy sets and fuzzy logic. It is the backbone for Chap. 3 and provides the foundation upon which type-2 fuzzy sets and systems are built in later chapters. Its coverage includes: crisp sets, a short biography of Prof. Zadeh (the father of fuzzy sets and fuzzy logic), type-1 fuzzy set defined, linguistic variables, returning to linguistic variables from a numerical value of a membership function, set-theoretic operations for crisp and type-1 fuzzy sets, crisp and fuzzy relations and compositions on the same or different product spaces, compositions of a type-1 fuzzy set with a type-1 fuzzy relation, hedges, the Extension Principle (which is about functions of fuzzy sets), α-cuts (which are a powerful way to represent a type-1 fuzzy set in terms of intervals), functions of type-1 fuzzy sets computed by using α-cuts, multivariable membership functions and Cartesian products, crisp logic, going from crisp logic to fuzzy logic, Mamdani (engineering) implications, some final remarks, and an appendix about properties/laws of type-1 fuzzy sets. To illustrate this chapter's important concepts 35 examples are used.

Chapter 3 explores many aspects of the Fig. 1.2 fuzzy system when all fuzzy sets are type-1. It provides a very comprehensive and unified description of the two major kinds of type-1 fuzzy systems that are widely used in real-world applications —Mamdani and TSK fuzzy systems. Its coverage includes: rules, singleton and non-singleton fuzzifiers, input–output formulas for the fuzzy inference engine, first- and second-order rule partitions, the effects of the two kind of fuzzifiers on the input–output formulas, combining or not combining fired rule output sets on the way to defuzzification, centroid, height and center-of-sets defuzzifiers, fuzzy basis functions (which provide a mathematical description of a fuzzy system from its input to its output), remarks and insights about a type-1 fuzzy system (including layered architecture interpretations for it, universal approximation by it, continuity of it, rule explosion and some ways to control it, and rule interpretability for it). To illustrate this chapter's important concepts 18 examples are used and there is also a comprehensive numerical example in Sect. 3.7 that is continued in later chapters. Chapter 9 builds upon the material that is in this chapter.

Chapter 4 focuses first on what exactly "design of a type-1 fuzzy system" means, and then provides a tabular way for making the choices that are needed in order to fully specify such a system. It introduces two approaches to design, the partially dependent approach and the totally independent approach, and then describes six design methods for designing a type-1 fuzzy system, namely: one-pass, least squares, derivative-based, SVD-QR, derivative-free, and iterative. It also introduces and covers the three case studies of forecasting of time series, knowledge mining using surveys, and fuzzy logic control (all of which are reexamined in Chap. 10), as well as the applications of forecasting of compressed video traffic, and rule-based classification of video traffic. To illustrate this chapter's important concepts 12 examples are used.

Chapter 5 examines the kinds of uncertainties that motivate the use of type-2 fuzzy sets and systems. Its coverage includes: general discussions about the occurrence, causes, and nature of uncertainty, uncertainties and sets, uncertainties in a fuzzy system, and collecting word data from a group of subjects to demonstrate that *words mean different things to different people*. This chapter demonstrates that uncertainty is a commodity that can be used to control the rule explosion that is so common in a fuzzy system.

Chapter 6 formally introduces type-2 fuzzy sets and its terminology, and is the backbone for the rest of the book. Its coverage includes: the concept of a type-2 fuzzy set, definitions of and associated concepts for general type-2 fuzzy sets and interval type-2 fuzzy sets, examples of two popular footprints of uncertainty, interval type-2 fuzzy numbers, a hierarchy of different kinds of type-2 fuzzy sets, three very useful mathematical representations of type-2 fuzzy sets—vertical-slice, wavy-slice, and horizontal-slice, discussions about which mathematical representations are the most useful for optimal designs of type-2 fuzzy systems, how to represent non-type-2 fuzzy sets as type-2 fuzzy sets, returning to linguistic labels for type-2 fuzzy sets, and multivariable membership functions. To illustrate this chapter's important concepts 24 examples are used.

Chapter 7 explains how to work with type-2 fuzzy sets. Most of its topics are needed in the rest of this book. Its coverage includes: set-theoretic operations (union, intersection and complement) for general type-2 fuzzy sets computed by using the Extension Principle or by using horizontal slices, set-theoretic operations for interval type-2 fuzzy sets, type-2 relations and compositions on the same or different product spaces, compositions of a type-2 fuzzy set with a type-2 relation, type-2 hedges, Extension Principle for interval type-2 and general type-2 fuzzy sets, functions of general type-2 fuzzy sets computed by using horizontal slices, Cartesian product of type-2 fuzzy sets, implications, an appendix about the properties/laws of type-2 fuzzy sets, and an appendix that has detailed proofs of many of the chapter's theorems. More than 25 examples are used to illustrate this chapter's important concepts.

Chapter 8 introduces *type-reduction* that lets a type-2 fuzzy set be projected into a type-1 fuzzy set and is often used in a type-2 fuzzy system as a first step in going from a type-2 fuzzy set to a number. This chapter's coverage includes: the interval weighted average (IWA), because it is the basic building block for type-reduction, three algorithms (KM, EKM and EIASC) for computing the IWA, centroid type-reduction for interval type-2 fuzzy sets and systems, height and center-of-sets type-reduction for interval type-2 fuzzy systems, centroid type-reduction for general type-2 fuzzy sets and systems, height and center-of-sets type-reduction for general type-2 fuzzy systems, an appendix that presents (for historical reasons) the early approach to type-reduction, and an appendix about the mathematical properties of the IWA, and about continuous algorithms for performing centroid type-reduction. To illustrate the chapter's important concepts 14 examples are used.

Chapter 9 explores many aspects of the Fig. 1.2 fuzzy system when all of the fuzzy sets are interval type-2. As was done for type-1 fuzzy systems, it provides a very comprehensive and unified description of the two major kinds of interval type-2 (IT2) fuzzy systems that are widely used in real-world applications—IT2 Mamdani and TSK fuzzy systems. Importantly, it also distinguishes between IT2 fuzzy systems that include type-reduction followed by defuzzification and those that bypass type-reduction and use direct defuzzification. The coverage of this chapter includes: IT2 rules, three kinds of fuzzifiers (singleton, type-1 non-singleton and IT2 non-singleton), input–output formulas for the fuzzy inference engine (valid even for general type-2 fuzzy sets), the effects of the three kind of fuzzifiers on the input–output formulas (valid for IT2 fuzzy sets), IT2 first- and second-order rule partitions, combining or not combining fired rule output sets on the way to defuzzification, type-reduction (centroid, height, and center-of-sets) + defuzzification for an IT2 Mamdani fuzzy system, type-reduction + defuzzification for four kinds of IT2 TSK fuzzy systems, novelty partitions, approximate type-reduction and defuzzification (Wu–Mendel Uncertainty Bounds), direct defuzzification

(Nie-Tan and Biglarbegian–Melek–Mendel), IT2 fuzzy basis functions (which provide a mathematical description of an IT2 fuzzy system from its input to its output), remarks and insights about an IT2 fuzzy system (including layered architecture interpretations for it, fundamental differences between type-1 and IT2 fuzzy systems, universal approximation by it, continuity of it, rule explosion and some ways to control it, and rule interpretability for it), and historical notes. To illustrate the chapter's important concepts 17 examples are used and there is also a comprehensive numerical example in Sects. 9.7 and 9.11.

Chapter 10 is the IT2 version of Chapter 4. It focuses first on what exactly "design of an IT2 fuzzy system" means, and then provides a tabular way for making the choices that are needed in order to fully specify an IT2 fuzzy system. It introduces two approaches to design, the partially dependent approach and the totally independent approach, but this time for singleton, T1 non-singleton, and IT2 non-singleton IT2 fuzzy systems. It then describes the extension of the six design methods that were covered for type-1 fuzzy systems in Chap. 4 to IT2 fuzzy systems, namely: IT2 WM, least squares, derivative-based, SVD-QR, derivative-free, and iterative. It continues the three Chap. 4 case studies of forecasting of time series, knowledge mining using surveys, and fuzzy logic control, and also continues the Chap. 4 applications of forecasting of compressed video traffic using IT2 Mamdani and TSK fuzzy systems, and IT2 rule-based classification of video traffic. The application of equalization of time-varying nonlinear digital communication channels is also covered. To illustrate the chapter's important concepts 13 examples are used.

Chapter 11 explores many aspects of the Fig. 1.2 fuzzy system when all of the fuzzy sets are general type-2 (GT2). As was done for interval type-2 fuzzy systems, it provides a very comprehensive and unified description of the two major kinds of general type-2 fuzzy systems that may be used in real-world applications—GT2 Mamdani and GT2 TSK fuzzy systems. Importantly, it also distinguishes between GT2 fuzzy systems that include type-reduction followed by defuzzification and those that bypass type-reduction and use direct defuzzification. The coverage of this chapter focuses on singleton fuzzification and the use of the horizontal-slice representation of a GT2 FS, and includes: GT2 rules, horizontal-slice formulas for firing sets and fired rules output sets, horizontal-slice first- and second-order rule partitions, combining or not combining fired rule output sets on the way to defuzzification, horizontal-slice type-reduction (centroid and center-of-sets) for horizontal-slice GT2 Mamdani and TSK fuzzy systems, defuzzification (this is where horizontal slices are aggregated), a summary of the computational steps for two horizontal-slice Mamdani and two horizontal-slice TSK GT2 fuzzy systems, horizontal-slice versions of the NT and BMM direct defuzzification methods, GT2 fuzzy basis functions (which provide a mathematical description of a GT2 fuzzy system from its input to its output), remarks and insights about a GT2 fuzzy system, what exactly "design of a GT2 fuzzy system" means as well as a tabular way for making the choices that are needed to fully specify a GT2 fuzzy system, two approaches to design—the partially dependent approach and the totally independent approach (but only for singleton GT2 fuzzy systems), requirements that need to be met in the study of real-world applications of GT2 fuzzy systems, and a case study of GT2 fuzzy logic control. To illustrate the important concepts ten examples are used and there is also a comprehensive numerical example in Sects. 11.9 and 11.11.

1.8 Applicability Outside of Rule-Based Fuzzy Systems

Although this book is about rule-based fuzzy systems, much of Chaps 2, 6, 7 and 8 are also applicable to non-rule-based applications of type-1 and type-2 fuzzy sets. Such applications are left to the reader to explore.

1.9 Computation

As of the writing of this book (2016), the following sources are available[10] for software that can be used to implement much of what is in this book:

1. A free open-source[11] MATLAB®/SIMULINK® Toolbox for interval type-2 fuzzy logic systems. It can be accessed at: http://web.itu.edu.tr/kumbasart/type2fuzzy.htm. Its developers are Ahmet Taskin and Tufan Kumbasar.
2. Functions for Interval Type-2 Fuzzy Logic Systems: It is MATLAB® based, free and can be accessed at: https://www.mathworks.com/matlabcentral/fileexchange/29006-functions-for-interval-type-2-fuzzy-logic-systems. Its developer is Dongrui Wu.
3. Juzzy, a free, open-source, Java-based library for the design and implementation of type-1, interval and general type-2 sets and system-based applications. It can be accessed at: http://lucidresearch.org/software. Its developer is Christian Wagner.
4. Type-2 fuzzy logic software (a collection of m-files for MATLAB® that includes m-files for type-1 fuzzy systems): It is free and can be accessed at: http://sipi.usc.edu/~mendel/ (go to Publications/Software/Software/I agree to these conditions). Its developers are: Nilesh Karnik, Qilian Liang, Feilong Liu, Dongrui Wu, Jhiin Joo, Minshen Hao, and Jerry M. Mendel.

---

[10]There also is other proprietary software that is being used by researchers, but, even though it is used, mentioned, described, and referenced in articles, it is not available to others.

[11]MATLAB and SIMULINK are registered trademarks of The MathWorks.

References

Aliev, R. A., and B. G. Guirimov. 2014. *Type-2 fuzzy neural networks and their applications*. Heidelberg: Springer.

Bustince, H., E. Barrenechea, M. Pagola, J. Fernandez, Z. Xu, B. Bedregal, J. Montero, H. Hagras, F. Herrera, and B. De Baets. 2016. A historical account of types of fuzzy sets and their relationships. *IEEE Transactions on Fuzzy Systems* 24: 179–194.

Castillo, O. 2012. *Type-2 fuzzy logic in intelligent control applications*, vol. 272 in studies in fuzziness and soft computer, Heidelberg: Springer.

Castillo, O., and P. Melin. 2008. *Type-2 fuzzy logic theory and applications*. Berlin: Springer.

Castillo, O., and P. Melin. 2012. *Recent advances in Interval type-2 fuzzy systems*. Springer Briefs in Applied Sciences Computational Intelligence. Heidelberg: Springer.

Castillo, O., and P. Melin. 2014. A review on interval type-2 fuzzy logic applications in intelligent control. *Information Sciences* 279: 615–631.

Cervantes, L., and O. Castillo. 2016. *Hierarchical type-2 fuzzy aggregation of fuzzy controllers*. Springer briefs in computational intelligence, 69. Heidelberg: Springer.

Chaneau, J. L., M. Gunaratne and A. G. Altschaeffl. 1987. An application of type-2 sets to decision making in engineering. In *Analysis of fuzzy information, vol. II: Artificial intelligence and decision systems*, ed. J. Bezdek. Boca Raton: CRC press.

Chen, Z., Aghakhani, A., Man, J., and S. Dick. 2010. ANCFIS: Aneurofuzzy architecture employing complex fuzzy sets. *IEEE Transactions on Fuzzy Systems* 19: 305–322.

Chiang, D.A., L.-R. Chow, and N.-C. Hsien. 1997. Fuzzy information in extended fuzzy relational databases. *Fuzzy Sets and Systems* 92: 1–20.

Dereli, T., Baykasoglu, A., Altun, K., Durmusoglu, A., and I. Burkhan Turksen. 2011. Industrial applications of type-2 fuzzy sets and systems: A concise review. *Computers in Industry* 62: 125–137.

Dick, S. 2005. Toward complex fuzzy logic. *IEEE Transactions on Fuzzy Systems* 13: 405–414.

Dubois, D., and H. Prade. 1978. Operations on fuzzy numbers. *International Journal Systems Science* 9: 613–626.

Dubois, D., and H. Prade. 1979. Operations in a fuzzy-valued logic. *Information and Control* 43: 224–240.

Dubois, D., and H. Prade. 1980. *Fuzzy sets and systems: Theory and applications*. NY: Academic Press.

Dziech, A., and M.B. Gorzalczany. 1987. Decision making in signal transmission problems with interval-valued fuzzy sets. *Fuzzy Sets and Systems* 23 (2): 191–203.

Gorzalczany, M.B. 1987. A method of inference in approximate reasoning based on interval-valued fuzzy sets. *Fuzzy Sets and Systems* 21: 1–17.

Gorzalczany, M.B. 1988. Interval-valued fuzzy controller based on verbal model of object. *Fuzzy Sets and Systems* 28: 45–53.

Gorzalczany, M.B. 1989a. Interval-valued fuzzy inference involving uncertain (inconsistent) conditional propositions. *Fuzzy Sets and Systems* 29: 235–240.

Gorzalczany, M.B. 1989b. An interval-valued fuzzy inference method. Some basic properties. *Fuzzy Sets and Systems* 31: 243–251.

Gratten-Guiness, I. 1976. Fuzzy membership mapped onto interval and many-valued quantities. *Zeitschrift für Mathematische Logik and Grundladen der Mathematik* 23: 149–160.

Hagras, H. 2007. Type-2 FLCs: A new generation of fuzzy controllers. *IEEE Computational Intelligence Magazine* 2: 30–43.

Hagras, H., and C. Wagner. 2012. Towards the wide spread use of type-2 fuzzy logic systems in real world applications. *IEEE Computational Intelligence Magazine* 7 (3): 14–24.

Hagras, H., D. Alghazzawi, and G. Aldabbagh. 2015. Employing type-2 fuzzy logic systems in the efforts to realize ambient intelligent environments. *IEEE Computational Intelligence Magazine* 10 (1): 44–51.

Hisdal, E. 1981. The IF–THEN ELSE statement and interval-values fuzzy sets of higher type. *International Journal Man-Machine Studies* 15: 385–455.

Izumi, K., Tanaka, H., and K. Asai. 1983. Resolution of composite fuzzy relational equations of type 2. *Transactions of the Institute of Electronics and Communication Engineers of Japan (in Japanese), Part D*. J66D: 1107–1113.

Jahn, K.U. 1075. Intervall-wertige mengen. *Mathematische Nachrichten* 68: 115–132.

Jang, J.-S. R. and C-T. Sun. 1995. Neuro-fuzzy modeling and control. *IEEE Proceedings* 83: 378–406.

Jang, J.-S. R., Sun, C-T., and E. Mizutani. 1997. *Neuro-fuzzy and soft-computing*. Upper Saddle River: Prentice-Hall.

John, R. I., Type-2 inferencing and community transport scheduling, *Proceedings of the fourth European congress on intelligent techniques and soft computing*, *EUFIT'96*, pp. 1369–1372, Aachen, Germany, Sept. 1996.

John, R., and S. Coupland. 2007. Type-2 fuzzy logic: A historical view. *IEEE Computational Intelligence Magazine* 2: 57–62.

John, R. I., P. R. Innocent and M. R. Barnes. 1997. Type 2 fuzzy sets and neuro-fuzzy clustering or radiographic tibia images. In *Proceedings of sixth international conference on fuzzy systems* 1375–1380, Barcelona, Spain; also, in *Proceedings of IEEE international conference on fuzzy systems*. 1373–1376, Anchorage 1998.

Karnik, N. N., and J. M. Mendel. 1998a. Introduction to type-2 fuzzy logic systems. In *Proceedings 1998 IEEE fuzzy conference* pp. 915–920, Anchorage, AK.

Karnik, N. N., and J. M. Mendel. 1998b. *An introduction to type-2 fuzzy logic systems*. *USC-SIPI Report #418, University of Southern California*, Los Angeles, CA This can be Accessed at: http://sipi.usc.edu/research; then choose "sipi technical reports/418‶.

Karnik, N. N., and J. M. Mendel. 1998c. Type-2 fuzzy logic systems: Type-reduction. In *Proceedings IEEE conference on systems, man and cybernetics*, 2046–2051, San Diego, CA.

Karnik, N.N., and J.M. Mendel. 1999a. Applications of type-2 fuzzy logic systems to forecasting of time-series. *Information Sciences* 120: 89–111.

Karnik, N. N. and J. M. Mendel. 1999b. Applications of type-2 fuzzy logic systems: handling the uncertainty associated with surveys. In *Proceedings FUZZ-IEEE'99*, Seoul, Korea.

Karnik, N.N., and J.M. Mendel. 2001a. Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems* 122: 327–348.

Karnik, N.N., and J.M. Mendel. 2001b. Centroid of a type-2 fuzzy set. *Information Sciences* 132: 195–220.

Karnik, N.N., J.M. Mendel, and Q. Liang. 1999. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 7: 643–658.

Klir, G.J., and B. Yuan. 1995. *Fuzzy sets and fuzzy logic: Theory and applications*. Upper Saddle River: Prentice Hall.

Liang, Q., and J. M. Mendel. 1999. an introduction to type-2 tsk fuzzy logic systems. In *Proceedings FUZZ-IEEE'99*, Seoul, Korea.

Liang, Q., and J. M. Mendel. 2000a. Interval type-2 fuzzy logic systems. In *Proeedings. FUZZ-IEEE '00*, San Antonio, TX.

Liang, Q., and J. M. Mendel. 2000b. Decision feedback equalizer for nonlinear time-varying channels using type-2 fuzzy adaptive filters. In *Proceedings FUZZ-IEEE'00*, San Antonio, TX.

Liang, Q., and J.M. Mendel. 2000c. Interval type-2 fuzzy logic systems: Theory and design. *IEEE Transactions on Fuzzy Systems* 8: 535–550.

Liang, Q., and J.M. Mendel. 2000d. Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters. *IEEE Transactions on Fuzzy Systems* 8: 551–563.

Liang, Q., and J.M. Mendel. 2000e. Designing interval type-2 fuzzy logic systems using an SVD–QR method: Rule reduction. *International Journal of Intelligent Systems* 15: 939–957.

Liang, Q., and J. M. Mendel. 2000f. Overcoming time-varying co-channel interference using type-2 fuzzy adaptive filter. *IEEE Transactions on Circuits and Systems* 47: 1419–1428.

Liang, Q., and J. M. Mendel. 2000g. MPEG VBR video traffic modeling and classification using fuzzy techniques. *IEEE Transactions on Fuzzy Systems*

Liang, Q., and J. M. Mendel. 2001. Modeling MPEG VBR video traffic using type-2 fuzzy logic systems. In *Granular computing: An emerging paradigm*, ed. W. Pedrycz. Heidelberg: Springer.

Liang, Q., Karnik, N. N., and J. M. Mendel. 2000. Connection admission control in atm networks using survey-based type-2 fuzzy logic systems. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 30: 329–339.

Mamdani, E.H. 1974. Applications of fuzzy algorithms for simple dynamic plant. *Proceedings of the IEEE* 121: 1585–1588.

Mamdani, E.H., and S. Assilian. 1975. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Machine Studies* 7: 1–13.

McNeill, D., and P. Freilberger. 1992. *Fuzzy logic: The discovery of a revolutionary computer technology and how it is changing our world*. New York: Simon and Schuster.

Melin, P. 2012. *Modular neural networks and type-2 fuzzy systems for pattern recognition*. In Studies in computational intelligence. 389. Heidelberg: Springer.

Mendel, J.M. 1995. Fuzzy logic systems for engineering: A tutorial. *IEEE Proceedings* 83: 345–377.

Mendel, J.M. 2000. Uncertainty, fuzzy logic, and signal processing. *Signal Proceedings Journal* 80: 913–933.

Mendel, J.M. 2001. *Introduction to rule-based fuzzy logic systems*. Upper Saddle River: Prentice-Hall.

Mendel, J.M. 2007. Advances in type-2 fuzzy sets and systems. *Information Sciences* 177: 84–110.

Mendel, J.M. 2010. Type-2 fuzzy sets—A tribal parody. *IEEE Computational Intelligence Magazine* 5: 24–27.

Mendel, J.M., H. Hagras, W.-W. Tan, W.W. Melek, and H. Ying. 2014. *Introduction to type-2 fuzzy logic control*. Hoboken: Wiley and IEEE Press.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: Aiding people in making subjective judgments*. Hoboken: Wiley and IEEE Press.

Mizumoto, M., and K. Tanaka. 1976. Some properties of fuzzy sets of type-2. *Information and Control* 31: 312–340.

Mizumoto, M., and K. Tanaka. 1981. Fuzzy sets of type-2 under algebraic product and algebraic sum. *Fuzzy Sets and Systems* 5: 277–290.

Nieminen, J. 1977. On the algebraic structure of fuzzy sets of type-2. *Kybernetica* 13, 4.

Ross, T. 2004. *Fuzzy logic for engineering applications*, 2nd ed. UK: Wiley.

Sadeghian, A., J.M. Mendel, and H. Tahayori. 2013. *Advances in type-2 fuzzy sets and systems*. New York, NY: Springer.

Sambuc, R. 1975. *Function Φ-flous, application a l'aide au diagnostic en pathologie thyroidienne*. These de Doctorat en Medicine: University of Marseille.

Sugeno, M. 1983. Fuzzy theory, III. *Journal of the Society of Instrument and Control Engineers* (in Japanese) 22: 454–458.

Türksen, I.B. 1986. Interval valued fuzzy sets based on normal forms. *Fuzzy Sets and Systems* 20: 191–210.

Türksen, I.B. 1992. Interval-valued fuzzy sets and 'compensatory and'. *Fuzzy Sets and Systems* 51: 295–307.

Türksen, I. B., and D. D. Yao. 1984. Representation of connectives in fuzzy reasoning: The view through normal forms. *IEEE Transactions on Systems, Man and Cybernetics* 14: 191–210.

Türksen, I.B., and Z. Zhong. 1990. An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets. *Fuzzy Sets and Systems* 34: 323–346.

Wagenknecht, M., and K. Hartmann. 1988. Application of fuzzy sets of type-2 to the solution of fuzzy equation systems. *Fuzzy Sets and Systems* 25: 183–190.

Wang, L.-X. 1997. *A course in fuzzy systems and control*. Upper Saddle River: Prentice-Hall.

Wu, K.C. 1996. Fuzzy interval control of mobile robots. *Computers & Electrical Engineering* 22: 211–229.

Yager, R.R. 1980. Fuzzy subsets of type ii in decisions. *Journal of Cybernetics* 10: 137–159.

Yager, R.R., and D.P. Filev. 1994. *Essentials of fuzzy modeling and control*. New York: Wiley.

Yen, J., and R. Langari. 1999. *Fuzzy logic: Intelligence, control, and information*. Upper Saddle River: Prentice-Hall.

Zadeh, L.A. 1965. Fuzzy sets. *Information and control* 8: 338–353.

Zadeh, L.A. 1975. The concept of a linguistic variable and its application to approximate reasoning–1. *Information Sciences* 8: 199–249.

Chapter 2
Type-1 Fuzzy Sets and Fuzzy Logic

2.1 Crisp Sets

Recall that a *set* A in a universe of discourse X (which provides the set of allowable values for a variable) can be defined by listing all of its members or by identifying the elements $x \subset A$. One way to do the latter is to specify a condition or conditions for which $x \subset A$; thus, A can be defined as $A = \{x | x \text{ meets some condition(s)}\}$. Alternatively, one can introduce a zero-one *membership function* (MF) (also called a characteristic function, discrimination function, or indicator function) for A, denoted $\mu_A(x)$, such that

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

Set A (which can also be treated as a subset of X) is mathematically equivalent to its MF $\mu_A(x)$ in the sense that knowing $\mu_A(x)$ is the same as knowing A itself. In order to distinguish between a set and a fuzzy set, the former will be referred to as a "crisp set."

Example 2.1 (Mendel 1995a) Consider the set of all automobiles in New York City; this is X. The elements of X are individual cars; but, there are many different types of subsets that can be established for X, including the three that are depicted in Fig. 2.1. Either a car has or does not have six cylinders. This is a very crisp requirement. Hence, if your car has four cylinders, its MF value (i.e., membership grade) for the subset of four cylinder cars is unity, whereas its membership grades for the subsets of six cylinder or eight cylinder cars are zero.

Example 2.2 (Mendel 2015) Suppose that the domain of x is partitioned into five regions, and one knows exactly where the dividing line is between each region,

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_2

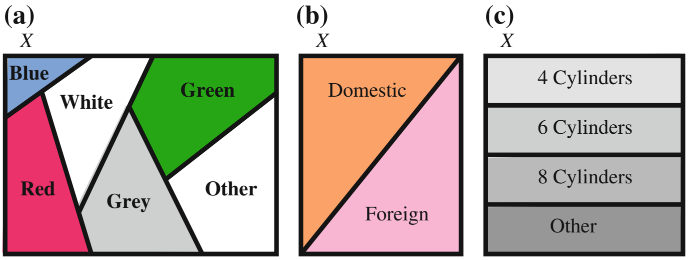
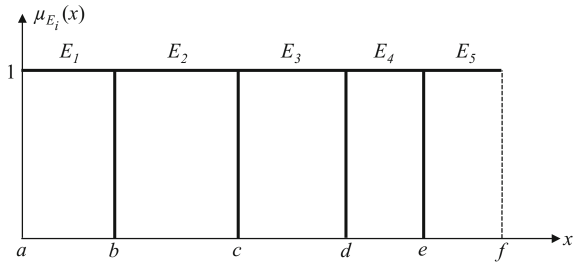


Fig. 2.1 Partitioning of the set of all automobiles in New York City into subsets by **a** color, **b** domestic or foreign, and **c** number of cylinders (Mendel 1995a © 1995, IEEE)

Fig. 2.2 Interpreting crisp sets as crisp partitions (Mendel 2015 © Springer 2015)



so one is in the situation that is depicted in Fig. 2.2, where no uncertainty exists about $x = b, c, d, e$. Each of the intervals $[a, b], (b, c], (c, d], (d, e], (e,f]$ is a *crisp partition* (Definition 1.1), i.e. x is either in it (with membership value of 1) or not in it (with membership value of 0), and x cannot simultaneously be in more than one of these intervals. Each interval is associated with a crisp set that is described by a linguistic term, E_1, or E_2, or $\ldots$, or E_5, such as a level of temperature or pressure, and there is always a sharp jump from one set to another at $x = b, c, d, e$. As mentioned in connection with Fig. 1.1a, this crisp model serves us well in many situations, but it does not allow any uncertainty about $x = b, c, d, e$. A fuzzy set will allow for this, as shall be seen.

2.2 Type-1 Fuzzy Sets and Associated Concepts

This section provides the background that is needed to read Chaps. 3 and 4. To begin, a short section about the father of fuzzy sets and logic, Professor Lotfi A. Zadeh, is provided.

2.2.1 *Lotfi A. Zadeh*

Fuzzy sets[1] were invented around 1965 by Prof. Lotfi A. Zadeh, but why? In Zadeh (1973), he states:

> Essentially our contention is that the conventional quantitative techniques of system analysis are intrinsically unsuited for dealing with humanistic systems or, for that matter, any system whose complexity is comparable to that of humanistic systems. The basis for this contention rests on what might be called the *principle of incompatibility*. Stated informally, the essence of this principle is that as the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics (a corollary to this principle may be stated succinctly as, "The closer one looks at a real-world problem, the fuzzier becomes its solution."). It is in this sense that precise quantitative analyses of the behavior of humanistic systems are not likely to have much relevance to the real world societal, political, economic, and other types of problems which involve humans either as individuals or in groups.

Prof. Zadeh[2] (Fig. 2.3), born in Baku, Azerbaijan on February 4, 1921, and educated at Alborz College in Tehran, the University of Tehran, M.I.T. and Columbia University, spent most of his career at the University of California at Berkeley, after ten years at Columbia University. He was already a famous system theorist when in 1965 he published what has now become the seminal paper on fuzzy sets (Zadeh 1965). This paper, which, as of February 2017, has been cited in Google Scholar more than 69,500 times, and is the most highly cited paper in all of computer science, marked the beginning of a new direction; by introducing the concept of a fuzzy set, that is a class with un-sharp boundaries, he provided a basis for a qualitative approach to the analysis of complex systems in which linguistic rather than numerical variables are employed to describe system behavior and performance. In this way, a much better understanding of how to deal with uncertainty may be achieved, and better models of human reasoning may be constructed. Although his unorthodox ideas were initially met with some skepticism, they have gained wide acceptance in recent years and have found application in just about every field imaginable. He is now acknowledged to be the "Father of Fuzzy Sets and Fuzzy Logic."

---

[1]The English word "fuzzy" has a negative connotation when it used in a technical context. It may be okay to describe a soft teddy bear, a cuddly pet, or a peach but for it to be used for mathematics and its applications is a red flag. Prof. Zadeh was well aware of this but felt that in 1965 "fuzzy" was the best word for him to use for this kind of a set. I propose that, after more than 50 years, these sets be called *Zadehian* sets. I am not going to use my proposed replacement in this book, because, although I would like to do it, if I did almost no one would know what I was talking about.

[2]This short biographical sketch was taken mostly from Mendel (2007).

Fig. 2.3 Professsor Lotfi A. Zadeh, the Father of Fuzzy Sets and Fuzzy Logic. Photo taken at Mendel Symposium, at University of Southern California, May 2009

2.2.2 Type-1 Fuzzy Set Defined

Definition 2.1 A *type-1 fuzzy set*[3] A is (Aisbett et al. 2010) a set function on universe X (sometimes denoted D_A) into $[0,1]$, possibly constrained to belong to a family such as continuous functions, i.e. $\mu_A{:}X \to [0,1]$. The MF of A is denoted $\mu_A(x)$ and is called a *type-1 MF*, i.e.

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

in which $0 \le \mu_A(x) \le 1$. A can also be expressed in fuzzy set notation[4] for continuous universes X, as

---

[3]In order to distinguish among different fuzzy set models, what were originally called *fuzzy sets* are in this book called *type-1 fuzzy sets*. Beginning with Chap. 6, type-2 fuzzy sets are studied.

[4]*Fuzzy set notation* was introduced in Zadeh (1965) and has remained popular for more than 50 years, although many people find it somewhat strange and object to its use of symbols such as the integral and summation. Aisbett et al. (2010) distinguish between "fuzzy set notation" and "standard mathematical notation." In Definition 2.1, $\mu_A{:}X \to [0,1]$ is the description of a type-1 fuzzy set in standard mathematical notation. My own preference is to use each notation where it is useful.

$$A = \int_{x \in X} \mu_A(x)/x \tag{2.3}$$

where $\int$ denotes union over all $x \in X$, or for discrete universes X_d, as

$$A = \sum_{x \in X_d} \mu_A(x)/x \tag{2.4}$$

where $\sum$ denotes union over all $x \in X_d$. The slash in (2.3) and (2.4) associates the elements in X with their membership grades, where $\mu_A(x) > 0$. The value of $\mu_A(x)$ is called the *degree of membership*, or *membership grade*, of x in A. If $\mu_A(x) = 1$ or $\mu_A(x) = 0$ for all $x \in X$, then the fuzzy set A reduces to a crisp set.

$\mu_A(x)$ is also said to provide a *measure of the degree of similarity* of an element in X to the fuzzy set. Note that A can also be treated as a subset of X. Unlike a crisp set, that can be described in different ways (as is explained in Sect. 2.1), a fuzzy set can only be described by its MF.

Example 2.1 (Continued) (Mendel 1995a) Referring to the middle of Fig. 2.1, observe that cars can also be partitioned into the two subsets, domestic and foreign. But, a car can be viewed as "domestic" or "foreign" from different perspectives. One perspective is that a car is domestic if it carries the name of a U.S. auto manufacturer; otherwise it is foreign. There is nothing fuzzy about this perspective. Many people today, however, feel that the distinction between a domestic and foreign automobile is not as crisp as it once was, because many of the components for what one considers to be domestic cars (e.g., Fords, GMs, and Chryslers) are produced outside of the United States. Additionally, some "foreign" cars are manufactured in the United States. Consequently, one could think of the MFs for domestic and foreign cars looking like $\mu_D(x)$ and $\mu_F(x)$ depicted in Fig. 2.4. Observe that a specific car (located along the horizontal axis by determining the percentage of its parts made in the United States) exists in both subsets simultaneously—domestic cars and foreign cars—but to different degrees of membership. For example, if a car has 75% of its parts made in the United States, then[5] $\mu_D(75\%) = 0.90$ and $\mu_F(75\%) = 0.25$. Ultimately, one would describe such a car as domestic. In fact, when one does this, the subset is decided upon by choosing it to be associated with the maximum of $\mu_D(75\%) = 0.90$ and $\mu_F(75\%) = 0.25$.

The main point of this example is to demonstrate that in a fuzzy set an element can reside in more than one set to different degrees of similarity. This cannot occur in a crisp set.

---

[5]For fuzzy sets, there is absolutely no requirement that $\mu_D(x) + \mu_F(x) = 1$, even though some authors impose this (e.g., Ruspini 1969; Bezdek 1981). When the constraint that the sum of the fuzzy set memberships must add to 1 for $x \in X$ is imposed, the result is called a *fuzzy partition*. Fuzzy partitions are not used in this book because, in the opinion of this author, they impose unnecessary constraints on fuzzy set MFs, especially when MF parameters are optimized, as is commonly done in rule-based fuzzy systems.

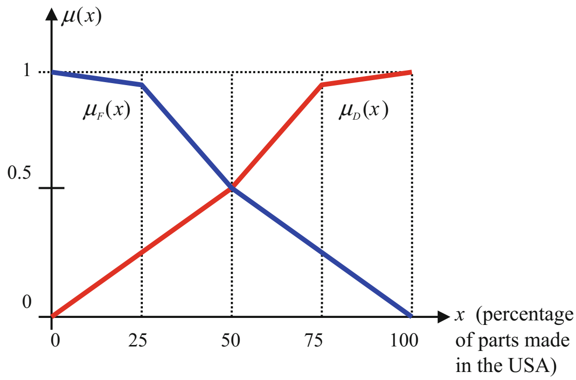


Fig. 2.4 MFs for domestic and foreign cars, based on the percentage of parts in the car made in the United States (Mendel 1995a © 1995, IEEE)

Note that describing a car by its color is also not a crisp description, because each color has different shades associated with it.

Definition 2.2 The *support* of a type-1 fuzzy set A is the crisp set of all points $x \in X$ such that $\mu_A(x) > 0$. A type-1 fuzzy set whose support is a single point in X with $\mu_A(x) = 1$ is called a (type-1) *fuzzy singleton*.

Definition 2.3 The *height* of a type-1 fuzzy set is the maximum MF value. A *normal* type-1 fuzzy set is one for which $\sup_{x \in X} \mu_A(x) = 1$, that is, its height equals 1.

Example 2.2 (Continued) (Mendel 2015) Referring to Fig. 2.2, suppose one now wants a model that allows for uncertainty about $x = b, c, d, e$, so that one is in the situation of Fig. 2.5, where in $[a, b_l]$ x resides only in E_1, whereas in $(b_l, b_r]$ x resides simultaneously in E_1 and E_2, but to different degrees, $\mu_{E_1}(x)$ and $\mu_{E_2}(x)$, respectively; in $(b_r, c_l]$ x resides only in E_2, whereas in $(c_l, c_r]$ x resides simultaneously in E_2 and E_3, but to different degrees, $\mu_{E_2}(x)$ and $\mu_{E_3}(x)$, respectively; etc. The MF $\mu_{E_i}(x)$ for E_i is no longer only 0 or 1, and MFs can overlap. So, a type-1 fuzzy set allows x to be partitioned using *overlapping partitions*, where one is absolutely certain about where the overlap begins and ends, i.e. as *first-order uncertainty partitions* (Definition 1.2), something that cannot be done by a crisp set. Overlapping partitions lead to smooth transitions from one set to another, which is very different from the sharp jumps that occur when crisp sets are used. As mentioned in connection with Fig. 1.1b, this fuzzy set model serves us well in many situations, but it does not allow for any uncertainty about the overlap. A type-2 fuzzy set will allow for this.

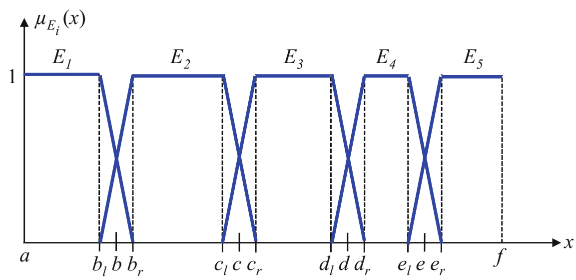


Fig. 2.5 Interpreting type-1 fuzzy sets as overlapping partitions (Mendel 2015 © Springer 2015)

Each of the five fuzzy sets in Fig. 2.5 is a normal type-1 fuzzy and the support of E_1 is $[a, b_r)$, the support of E_2 is $(b_l, c_r), \ldots$, and the support of E_5 is $(e_l, f]$.

Example 2.3 (Zimmerman 1991) Let F = integers close to 10; then, one choice for $\mu_F(x)$ is:

$$\mu_F(x) \equiv 0.1/7 + 0.5/8 + 0.8/9 + 1/10 + 0.8/11 + 0.5/12 + 0.1/13 \tag{2.5}$$

Three points to note from this MF are:

1. The integers for x not explicitly shown all have MFs equal to zero—by convention, such elements are not listed.
2. The values for the MFs were chosen by a specific individual; except for the unity membership value when $x = 10$, they can be modified based on one's own personal interpretation of the word "close," i.e. *words mean different things to different people*.
3. The MF is symmetric about $x = 10$, because there is no reason to believe that integers to the left of 10 are close to 10 in a different way than are integers to the right of 10; but again, other interpretations are possible.
4. F is a normal type-1 fuzzy set.
5. The fuzzy set F is an example of a *type-1 fuzzy number*, which will be defined formally in Sect. 2.2.3 (Definition 2.5).

Definition 2.4 A type-1 fuzzy set A is *convex* (Klir and Yuan 1995) if and only if

$$\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min[\mu_A(x_1), \mu_A(x_2)] \tag{2.6}$$

This can be interpreted as (Lin and Lee 1996): Take any two elements x_1 and x_2 in fuzzy set A; then the membership grade of all points between x_1 and x_2 must be greater than or equal to the minimum of $\mu_A(x_1)$ and $\mu_A(x_2)$. This will always occur when the MF of A is first monotonically non-decreasing and then monotonically non-increasing.[6] The five MFs in Fig. 2.5 are convex, whereas the two MFs in Fig. 2.4 are not.

Example 2.4 The MF of a convex type-1 fuzzy set A often satisfies the following structure:

$$\mu_A(x) = \begin{cases} g(x)|_{g(a)=0, g(b)=1}, & x \in [a, b] \\ 1, & x \in [b, c] \\ h(x)|_{h(c)=1, h(d)=0}, & x \in [c, d) \end{cases} \tag{2.7}$$

[6]In mathematics a real-value function $f(x)$ defined on an interval is called *convex* if the line segment between any two points on the graph of the function lies above or on the graph (e.g., a parabola). Why the fuzzy set A that satisfies (2.6) is called "convex" rather than "concave" is a bit mysterious. Maybe it is due to a concave function also being known in mathematics as a convex upwards, convex cap, or upper convex function.

where $g(x) \in [0,1]$ is monotonically non-decreasing and $h(x) \in [0,1]$ is monotonically non-increasing.

In rule-based applications of fuzzy logic, the MFs $\mu_A(x)$ are associated with linguistic terms that appear in the antecedents or consequents of rules, or in phrases (e.g., *foreign* cars).

Example 2.5 Some examples of rules and associated MFs (shown in brackets) are: (1) IF one is tracking a *large* target at one instant of time, THEN the target will not be *too far away* at the next instant of time $[\mu_{LARGE}(t), \mu_{TOO\text{-}FAR\text{-}AWAY}(x)]$; (2) IF the horizontal position is *medium positive* and the angular position is *small negative*, THEN the control angle is *large positive* $[\mu_{MEDIUM\text{-}POSITIVE}(x),\ \mu_{SMALL\text{-}NEGATIVE}(\theta),\ \mu_{LARGE\text{-}POSITIVE}(\phi)]$ and, (3) IF $y(t)$ is *close to 0.5*, THEN $f(y)$ is *close to zero* $[\mu_{CLOSE\text{-}TO\text{-}0.5}(y),\ \mu_{CLOSE\text{-}TO\text{-}ZERO}(f(y))]$.

The most commonly used shapes for MFs are triangular, trapezoidal, piecewise linear, Gaussian, and bell-shaped. MFs can either be chosen by the user arbitrarily, based on the user's experience (hence, the MFs for two users could be quite different depending upon their experiences, perspectives, cultures, etc.), or, they can be designed using optimization procedures, e.g., Horikawa et al. (1992), Jang (1992), Wang and Mendel (1992a, b).

The number of MFs is free to be chosen. Greater resolution is achieved by using more MFs at the price of greater computational complexity. MFs don't have to overlap; but one of the great strengths of fuzzy logic is that MFs can be made to overlap. This expresses the fact that "the glass can be partially full and partially empty at the same time." In this way (as will become clear in later chapters, e.g., Chap. 3) one is able to distribute decisions over more than one input class, which helps to make fuzzy logic systems robust.

The MF of a type-1 fuzzy set is specified exactly, which seems counter-intuitive for something that is supposed to be "fuzzy." This was one of the very early criticisms of a fuzzy set and is something that shall be returned to in Chap. 6 when type-2 fuzzy sets are studied.

2.2.3 Type-1 Fuzzy Numbers

When there is some uncertainty about a number (due, e.g., to measurement errors or linguistic uncertainty about it) it can be modeled as a fuzzy set, in which case it is called a *fuzzy number*. When the uncertainty is modeled using a type-1 fuzzy set it is called a *type-1 fuzzy number*. These numbers can be defined in different ways (e.g., Dubois and Prade 1980; Jang and Ralescu 2001; Klir and Yuan 1995; Wang 1997).

Definition 2.5 Let A be a fuzzy set in R. A is called a *type-1 fuzzy number* if: (i) A is normal, (ii) A is convex, and (iii) A has a bounded support.

It is tempting to do away with the requirement that A has a bounded support, but to do so makes no physical sense, since uncertainty about a real number should be finite. Regardless, it is not uncommon for an uncertain number to be modeled as a Gaussian type-1 fuzzy set that is centered about that number, and for this to be referred to as a "Gaussian fuzzy number." Strictly speaking, this designation is incorrect because the support for such a fuzzy set is unbounded. Occasionally, however, this designation is used even in this book, out of convenience, and because the author feels that its use will not confuse the reader.

Example 2.6 Formulas for triangle and trapezoidal type-1 fuzzy numbers are given in (2.8) and (2.9), respectively (see, also Table 2.3, in which alternate symbols are used for the parameters that define these fuzzy numbers):

$$\mu_A(x) = \mu_A(x; a, b, c) = \begin{cases} (x-a)/(b-a) & \text{if } a \le x < b \\ (c-x)/(c-b) & \text{if } b \le x \le c \\ 0 & \text{if } x > c \text{ or } x < a \end{cases} \tag{2.8}$$

$$\mu_A(x) = \mu_A(x; a, b, c, d) = \begin{cases} (x-a)/(b-a) & \text{if } a \le x < b \\ 1 & \text{if } b \le x \le c \\ (d-x)/(d-c) & \text{if } c < x \le d \\ 0 & \text{if } x > d \text{ or } x < a \end{cases} \tag{2.9}$$

Note, also, that when $b = c$ in (2.7), the resulting fuzzy set is often called an *LR fuzzy number* (Dubois and Prade 1980).

Type-1 fuzzy numbers are sometimes used in a type-1 rule-based fuzzy system during the front-end fuzzification process. More will be said about this in Sect. 2.2.3 (Definition 3.5). The extension of a type-1 fuzzy number to an interval type-2 fuzzy number is described in Sect. 6.5.

Definition 2.6 A *type-1 interval fuzzy number A* is a type-1 fuzzy number for which $\mu_A(x) = 1$, $x \in [l, r]$.

These kinds of type-1 fuzzy numbers play an important role in interval type-2 fuzzy sets and systems.

2.2.4 Linguistic Variables

Zadeh (1975, p. 201) states:

> In retreating from precision in the face of overpowering complexity, it is natural to explore the use of what might be called linguistic variables, that is, variables whose values are not numbers but words or sentences in a natural or artificial language. The motivation for the use of words or sentences rather than numbers is that linguistic characterizations are, in general, less specific than numerical ones.

Definition 2.7 If a variable can take words in natural languages as its values, it is called a *linguistic variable*, where the words are characterized by fuzzy sets defined in the universe of discourse in which the variable is defined (Wang 1997). Each

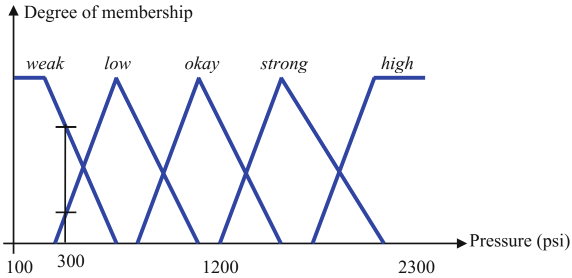


Fig. 2.6 MFs for T (pressure) = {*weak*, *low*, *okay*, *strong*, *high*}. The shapes of the MFs as well as their degree of overlap are quite arbitrary [©1992 IEEE. This figure has been taken from Cox (1992)]

linguistic variable (Klir and Yuan 1995; Zadeh 1973, 1975) is fully characterized by a quintuple (v, T, X, g, m) in which v is the name of the variable, T is the set of linguistic terms[7] of v that refer to a base variable whose values range over the universal set X, g is a syntactic rule for generating linguistic terms, and m is a semantic rule that assigns each linguistic term $t \in T$ its meaning, $m(t)$, which is a fuzzy set on X, that is, $m{:}T \to F(X)$, where $F(X)$ denotes the set of fuzzy sets of X, one fuzzy set for each $t \in T$. It is common to refer to v as the linguistic variable.

Example 2.7 Some examples of linguistic variables, v, are: Pressure, Horsepower, Acceleration, Production Rate, Developed Country, Industrial Country, Profitable Company, Institutional Veto Points, All-day School Systems, etc. Some examples of the set of linguistic terms, T, for these linguistic variables are[8]:

1. For Pressure, T = {*weak, low, okay, strong, high*}
2. For Horsepower, Acceleration and Production Rate, $T \triangleq$ {*very low, low, moderate, high, very high*}
3. For Developed or Industrial (Country) and Profitable (Company), $T \triangleq$ {*barely, hardly, somewhat, moderately, fully, extremely*}
4. For Institutional Veto Points and All-day School Systems, $T \triangleq$ {*none to very few, some, a moderate amount of, many, a large number of, a very large number of*}.

Observe that linguistic terms should make linguistic sense for its linguistic variable, which is where g in Definition 2.7 comes into play; so, for example, *somewhat acceleration* makes no linguistic sense nor does *very high all-day school systems*. Note, also, that *it is the elements of T that are treated as fuzzy sets,* and, of course, each of these fuzzy sets is described by a MF.

Figure 2.6 depicts the MFs for Pressure (Cox 1992) when its universe of discourse is X = [100 psi, 2300 psi]. One might interpret *weak* as a pressure below

[7]Although "term" means one or more words, it is quite common in the fuzzy set literature to see "word" used instead of "term," even when a term includes more than one word. In this book, "term" and "word" are also used interchangeably.

[8]Because some of the linguistic terms may be so similar to each other, it may not be necessary to use all of them. One usually chooses the linguistic terms so that their MFs overlap and cover X.

200 psi, *low* as a pressure close to 700 psi, *okay* as a pressure close to 1050 psi, *strong* as a pressure close to 1500 psi, and *high* as a pressure above 2200 psi. Measured values of pressure (x) lie along the pressure axis, and a vertical line from any value of pressure intersects, at most, two MFs. So, for example, $x = 300$ psi resides in the fuzzy sets *weak pressure* and *low pressure*, but to different degrees of similarity.

Zadeh (1999, p. 107) has used the word *perception* to describe the terms associated with linguistic variables. For example, he states:

> A fundamental difference between measurements and perceptions is that, in general, measurements are crisp numbers whereas perceptions are fuzzy numbers or, more generally, fuzzy granules, that is, clumps of objects in which the transition from membership to non-membership is gradual rather than abrupt.

Indeed, in Example 2.7, the terms *weak, low, okay, strong*, and *high* are perceptions about the level of pressure.

Example 2.8 Let X be the set of all men. The term "height" can mean different things to different people. Figure 2.7 depicts two sets of MFs for the set of terms {*short men, medium men, tall men*}. Clearly, the terms *short men*, *medium men*, and *tall men* will have a very different meaning to a professional basketball player than they will to most other people. This illustrates the fact that MFs can be quite context dependent.

The number of linguistic terms in T for a linguistic variable v will affect the calibration of the fuzzy sets, (i.e., the specification of its MFs). If, for example, only three linguistic terms are used to describe Height, namely {*short*, *medium*, *tall*}, then their fuzzy sets will look very different from their fuzzy sets when the following seven terms are used: {*very short*, *moderately short*, *short*, *medium*, *moderately tall*, *tall*, *very tall*}. This is because the terms *very short* and *moderately short* now appear before *short*, and *tall* is sandwiched between *moderately tall* and *very tall*. In many applications of rule-based fuzzy systems, the names that are given to the fuzzy sets are unimportant because interpretability of the rules is unimportant; however, there are other applications where interpretability of rules is very important. More is said about this in Sects. 3.9.5 and 9.13.6.

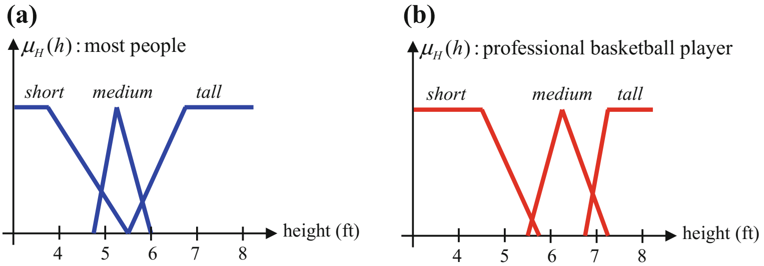


Fig. 2.7 MFs for T(height) = {*short men*, *medium men*, *tall men*}. **a** Most people's MFs, and **b** professional basketball player's MFs (Mendel 1995a © 1995, IEEE)

2.2.5 Returning to Linguistic Labels from Numerical Values of MFs

Sometimes it is necessary to go from MF numerical values for a variable to a linguistic description of that variable. This section examines how to do this for type-1 fuzzy sets.

Consider, for example, the linguistic variable temperature that has been decomposed into five terms {*very negative, medium negative, near zero, medium positive, very positive*}, and the situation depicted in Fig. 2.8 at $x = x'$. This value of x only generates a non-zero membership value in the fuzzy set F_4 = *medium positive*; hence, $x = x'$ can be described linguistically, without any ambiguity, as "*medium positive*."

The situation at $x = x''$ is different, because this value of x generates a non-zero membership value in two fuzzy sets F_4 = *medium positive* and F_5 = *very positive*. It would be very awkward to speak of x'' as "being *medium positive* to degree $\mu_{F_4}(x'')$ and *very positive* to degree $\mu_{F_5}(x'')$." People just don't communicate this way. Instead, $\mu_{F_4}(x'')$ and $\mu_{F_5}(x'')$ are compared to see which is larger, and then x'' is assigned to the set associated with the larger value; hence, in this example, x'' would be described as being "*medium positive.*"

What has just been explained can be descried formally as follows. Given P fuzzy sets F_i with MFs $\mu_{F_i}(x)\ (i = 1, \ldots, p)$. When $x = x'$, evaluate all p MFs at this point, and then compute $\max[\mu_{F_1}(x'), \mu_{F_2}(x'), \ldots, \mu_{F_p}(x')] \equiv \mu_{F_m}(x')$. Let $L(x')$ denote the linguistic label associated with x'. Then, $L(x') \equiv F_m$, i.e.,

$$L(x') = \arg\max_{\forall F_i}\left[\mu_{F_1}(x'), \mu_{F_2}(x'), \ldots, \mu_{F_p}(x')\right] \tag{2.10}$$

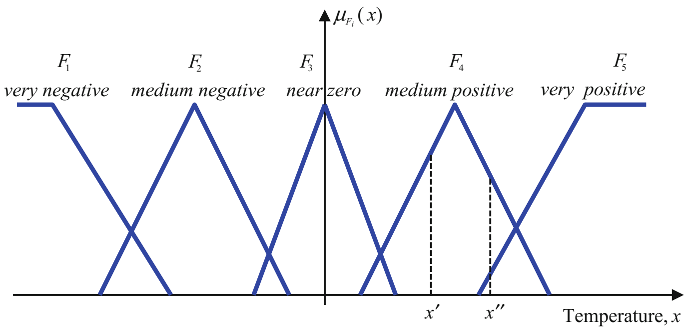


Fig. 2.8 Returning to a linguistic label for type-1 fuzzy sets

2.3 Set Theoretic Operations for Crisp Sets

Now that fuzzy sets have been defined, what can one do with them? The same question could be asked about crisp sets, and one knows that there are lots of things that can be done with them; hence, it is expected that analogous things can be done with fuzzy sets. To begin, the elementary crisp-set operations of union, intersection, and complement are briefly reviewed.

Let A and B be two subsets of X. The *union* of A and B, denoted $A \cup B$, contains all of the elements in either A or B, i.e.,

$$\mu_{A \cup B}(x) = \begin{cases} 1 & \text{if } x \in A \text{ or } x \in B \\ 0 & \text{if } x \notin A \text{ and } x \notin B \end{cases} \tag{2.11}$$

The *intersection* of A and B, denoted $A \cap B$, contains all of the elements that are simultaneously in A and B, i.e.,

$$\mu_{A \cap B}(x) = \begin{cases} 1 & \text{if } x \in A \text{ and } x \in B \\ 0 & \text{if } x \notin A \text{ or } x \notin B \end{cases} \tag{2.12}$$

Let $\bar{A}$ denote the *complement* of A; it contains all the elements that are not in A, i.e.,

$$\mu_{\bar{A}}(x) = \begin{cases} 1 & \text{if } x \notin A \\ 0 & \text{if } x \in A \end{cases} \tag{2.13}$$

From these facts, it is easy to show that:

$$A \cup B \Rightarrow \mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \tag{2.14}$$

$$A \cap B \Rightarrow \mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \tag{2.15}$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{2.16}$$

Consider $\mu_{A \cup B}(x)$ for example. In this case, $x \in A$ or $x \in B$ means:

$$(\mu_A(x) = 1, \mu_B(x) = 1) \text{ or } (\mu_A(x) = 1, \mu_B(x) = 0) \text{ or } (\mu_A(x) = 0, \mu_B(x) = 1),$$

and, for each of these situations, $\max[\mu_A(x), \mu_B(x)] = 1$. Additionally, $x \notin A$ and $x \notin B$ means $(\mu_A(x) = 0, \mu_B(x) = 0)$ for which $\max[\mu_A(x), \mu_B(x)] = 0$. Consequently, $\max[\mu_A(x), \mu_B(x)]$ for $\forall x$ does provide the correct MF, given in (2.11), for union.

The formulas in (2.14)–(2.16), for $\mu_{A \cup B}(x)$, $\mu_{A \cap B}(x)$, and $\mu_{\bar{A}}(x)$, are very useful for proving other theoretical properties about crisp sets. Note, also, that the maximum and minimum are not the only ways to describe $\mu_{A \cup B}(x)$ and $\mu_{A \cap B}(x)$. While these formulas are not usually part of conventional set theory, they are essential to fuzzy set theory; however, as has just been demonstrated, they really do occur in conventional set theory. See, e.g., Klir and Folger (1988) and Yager and Filev (1994) for other ways to characterize these operations.

The crisp union and intersection operations satisfy many properties (see Table 2.8 in Appendix 1 to this chapter for an extensive list of these properties), including:

1. Commutative

$$A \cup B = B \cup A$$

2. Associative

$$A \cup B \cup C = (A \cup B) \cup C = A \cup (B \cup C)$$

3. Distributive

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \text{ and } A \cup (B \cap C) = (A \cup B) \cap (A \cup C).$$

These properties can be proved either by Venn diagrams or by means of the MF definition given in (2.1).

De Morgan's laws for crisp sets are:

$$\bullet\, \overline{A \cup B} = \bar{A} \cap \bar{B}$$
$$\bullet\, \overline{A \cap B} = \bar{A} \cup \bar{B}$$

These laws, which are also very useful in proving things about more complicated operations on sets, can also be proved either by Venn diagrams or by means of the MF definition given in (2.1).

The two fundamental (Aristotelian) laws of crisp set theory are:

1. *Law of Excluded Middle*: $A \cup \overline{A} = X$ (i.e., a set and its complement must comprise the universe of discourse).
2. *Law of Contradiction* $A \cap \overline{A} = \emptyset$ (i.e., an element can either be in its set or its complement; it cannot simultaneously be in both).

Fuzzy sets usually break these Aristotelian laws.

2.4 Set Theoretic Operations for Type-1 Fuzzy Sets

For fuzzy sets, union, intersection, and complement are defined in terms of their MFs. Let fuzzy sets A and B be described by their MFs $\mu_A(x)$ and $\mu_B(x)$. One definition of *fuzzy union* leads to the MF

$$\mu_{A\cup B}(x) = \max[\mu_A(x), \mu_B(x)] \tag{2.17}$$

and one definition of *fuzzy intersection* leads to the MF

$$\mu_{A\cap B}(x) = \min[\mu_A(x), \mu_B(x)] \tag{2.18}$$

Additionally, the MF for *fuzzy complement* is

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{2.19}$$

Obviously, these three definitions were motivated by their crisp counterparts in (2.14)–(2.16).

Example 2.9 (Mendel 1995a) In engineering the **damping ratio** is a dimensionless measure describing how oscillations in a system decay after a disturbance. Consider the fuzzy sets A = damping ratio x *considerably larger* than 0.5, and B = damping ratio x *approximately* 0.707. Note that damping ratio is a positive real number, i.e., its universe of discourse, X, is the positive real numbers $0 \le x \le 1$. Consequently, $A = \{(x, \mu_A(x)) | x \in X\}$ and $B = \{(x, \mu_B(x)) | x \in X\}$, where, for example, $\mu_A(x)$ and $\mu_B(x)$ are specified (by this author), as:

$$\mu_A(x) = \begin{cases} 0 & \text{if } 0 \le x \le 0.5 \\ \dfrac{5(x-0.5)^2}{1+(x-0.5)^2} & \text{if } 0.5 < x \le 1 \end{cases} \tag{2.20}$$

and

$$\mu_B(x) = \frac{1}{[1+(x-0.707)^4]} \quad 0 \le x \le 1 \tag{2.21}$$

Figure 2.9 depicts $\mu_A(x), \mu_B(x), \mu_{A\cup B}(x), \mu_{A\cap B}(x)$ and $\mu_{\bar{B}}(x)$. Observe, from Fig. 2.9a that $\mu_A(0.707) + \mu_B(0.707) > 1$ and from Fig. 2.9d, that the point $x = 0.5$ exists in both B and $\overline{B}$ simultaneously, but to different degrees, because $\mu_B(0.5) \neq 0$ and $\mu_{\bar{B}}(0.5) \neq 0$.

This example demonstrates that for fuzzy sets the classical Laws of Excluded Middle and Contradiction are broken, i.e., *for fuzzy sets*: $A \cup \overline{A} \neq X$ and $A \cap \overline{A} \neq \emptyset$. This has also been observed in Fig. 2.4 for the automobile Example 2.1 (continued). In fact, one of the ways to describe the difference between crisp set theory and fuzzy set theory is to explain that these two laws do not hold in fuzzy set theory.[9]

The maximum and minimum operators are not the only ones that could have been chosen to model fuzzy union and fuzzy intersection. Zadeh, in his pioneering first paper (Zadeh 1965), defined two operators each for fuzzy union and fuzzy intersection, namely:

[9]There is a small subset of type-1 fuzzy set theory that requires both of these laws to be satisfied. This work has had no impact on rule-based fuzzy systems and so it is not discussed in this book.

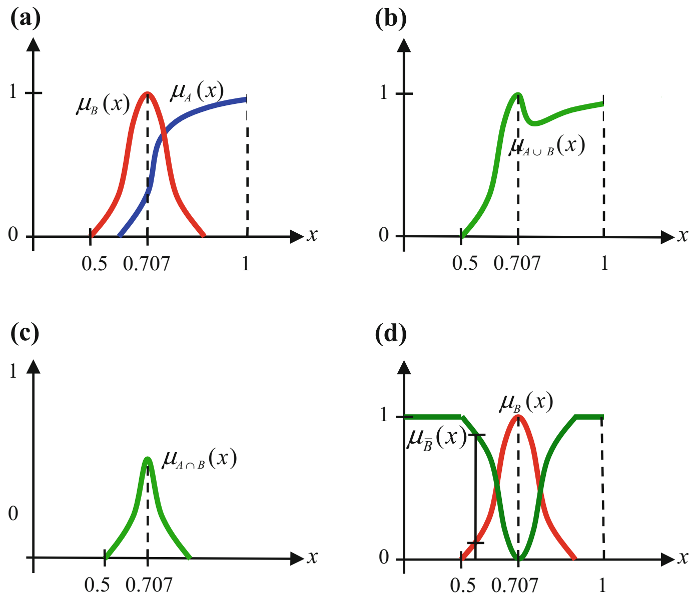


Fig. 2.9 MFs associated with A = damping ratio x *considerably larger* than 0.5, and B = damping ratio x *approximately* 0.707. **a** $\mu_A(x)$ and $\mu_B(x)$, **b** $\mu_{A\cup B}(x)$, **c** $\mu_{A\cap B}(x)$, and **d** $\mu_{\bar{B}}(x)$

1. Fuzzy union: maximum and algebraic sum, where for the latter

$$\mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) \tag{2.22}$$

2. Fuzzy intersection: minimum and algebraic product, where for the latter

$$\mu_{A\cap B}(x) = \mu_A(x)\mu_B(x) \tag{2.23}$$

Later, other operators that have an axiomatic basis (e.g., Klir and Yuan 1995) were introduced (in all cases, $x, y \in [0, 1]$):

1. **t-conorm operators**[10] for fuzzy union (also known as s-norm and denoted $\oplus$). The maximum and algebraic sums are t-conorms; some other examples of t-conorms are:
   - *Bounded sum*: $x \oplus y = \min(1, x + y)$

[10]The axiomatic basis for a *t-conorm* is, for *a, b, d* $\in$ [0,1]: (1) boundary condition, $s(a, 0) = a$; (2) monotonicity, $b \le d \Rightarrow s(a, b) \le s(a, d)$; (3) commutativity, $s(a, b) = s(b, a)$; and, (4) associativity, $s(a, s(b, d)) = s(s(a, b), d)$. Table 3.3 in Klir and Yuan (1995) lists 11 t-conorms.

- *Drastic sum*: $x \oplus y = \begin{cases} x & \text{if } y = 0 \\ y & \text{if } x = 0 \\ 1 & \text{otherwise} \end{cases}$

2. **t-norm operators**[11] for fuzzy intersection (denoted ★). The minimum and algebraic product are t-norms; some other examples of t-norms are:

   - *Bounded product*: $x \bigstar y = \max(0, x + y - 1)$
   - *Drastic product*: $x \bigstar y = \begin{cases} x & \text{if } y = 1 \\ y & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}$

There is even an axiomatic basis for the complement (denoted c) of a fuzzy set.[12] In engineering applications, most people use the fuzzy complement whose MF is given in (2.19).

As pointed out in Zimmerman (1991), dual pairs[13] of t-norms and t-conorms with respect to the fuzzy complement in (2.19) satisfy the following generalization of DeMorgan's laws (Bonissone and Decker 1986):

$$s[\mu_A(x), \mu_B(x)] = c\{t[c(\mu_A(x)), c(\mu_B(x))]\} \tag{2.24}$$

$$t[\mu_A(x), \mu_B(x)] = c\{s[c(\mu_A(x)), c(\mu_B(x))]\} \tag{2.25}$$

where $x \in X$. For example,

$$\max[\mu_A(x), \mu_B(x)] = 1 - \min[1 - \mu_A(x), 1 - \mu_B(x)] \tag{2.26}$$

and

$$\min[\mu_A(x), \mu_B(x)] = 1 - \max[1 - \mu_A(x), 1 - \mu_B(x)] \tag{2.27}$$

Note, also, that there are other ways of combining fuzzy sets, e.g., the *fuzzy and*, *fuzzy or*, *compensatory and*, and *compensatory or*; e.g., see Zimmerman (1991) and Yager and Filev (1994).

If at this point you are puzzled by all of the possible choices, a discussion about this is provided in Sect. 2.18.

---

[11]The axiomatic basis for a *t-norm* is, for *a, b, d* ∈ [0,1]: (1) boundary condition, $t(a, 1) = a$; (2) monotonicity, $b \le d \Rightarrow t(a, b) \le t(a, d)$; (3) commutativity, $t(a, b) = t(b, a)$; and, (4) associativity, $t(a, t(b, d)) = t(t(a, b), d)$. Table 3.2 in Klir and Yuan (1995) lists 11 t-norms.

[12]The axiomatic basis for a *fuzzy complement* is: (1) boundary conditions, $c(0) = 1$ and $c(1) = 0$, and (2) monotonicity, for all $a, b \in [0, 1]$, if $a \le b$ then $c(a) \ge c(b)$. There are also many fuzzy complements that additionally satisfy the involutive condition $c(c(a)) = a$.

[13]Some examples of dual pairs with respect to the fuzzy complement (2.19) are: min and max, and product and algebraic sum. See Klir and Yuan (1995, pp. 83–88) for discussions about and properties of dual pairs. Some of their Chap. 3 end-notes provide interesting historical remarks about the origins of t-norms and t-conorms.

Not only are the union, intersection, and complement performed with type-1 fuzzy sets, but sometimes other important set-theoretic operations are performed on them using well-known laws; e.g., commutative, associative, distributive, and De Morgan's laws (see Table 2.8 for a list of all the laws). For this book, an important question that needs to be answered is:

- Is it permissible to use a particular law for type-1 fuzzy sets under maximum t-conorm and either minimum or product t-norms?

Our focus is just on the maximum t-conorm and the minimum or product t-norms, because these are the most widely used ones in the fuzzy system's literature. The question must, of course, be re-examined if one uses other t-conorms and t-norms. Because the studies into the answers to this question, although important, are very technical, their details are presented in Appendix 1. Here, just the results are stated and some conclusions about them are drawn.

The aforementioned question has been very well studied for type-1 fuzzy sets (see Table 2.8) *For maximum t-conorm and minimum t-norm all laws are satisfied*; however, *for maximum t-conorm and product t-norm certain laws are not satisfied*. This means, therefore, that *one must be careful when using maximum t-conorm and product t-norm*. If, for example, the design of a maximum t-conorm and product t-norm type-1 fuzzy system involves the use of any of the violated laws it will be in error. *Fortunately, one usually does not have to use any of the violated laws in the creation and design of a type-1 fuzzy system*. The same cannot be said, in general, for other applications of type-1 fuzzy sets.

2.5 Crisp Relations and Compositions on the Same Product Space

According to Klir and Folger (1988, p. 65): *A crisp relation represents the presence or absence of association, interaction, or interconnectedness between the elements of two or more sets*. Here our attention is limited to relations between two sets U and V, i.e., to binary relations denoted $R(U,V)$. $U \times V$ denotes the Cartesian product of the two crisp sets U and V, i.e.,

$$U \times V = \{(u,v) | u \in U \text{ and } v \in V\} \tag{2.28}$$

$R(U,V)$ is a subset of $U \times V$.

Crisp relation $R(U,V)$ can be defined by the following MF:

$$\mu_R(u,v) = \begin{cases} 1 & \text{if and only if } (u,v) \in R(U,V) \\ 0 & \text{otherwise} \end{cases} \tag{2.29}$$

For binary relations defined over a Cartesian product whose elements come from a discrete universe of discourse, it is convenient to collect the MFs into a relational matrix whose elements are either zero or unity. An equivalent representation for a binary relation is a sagittal diagram, in which the sets U and V are each represented by a set of nodes in the diagram that are clearly distinguished from one another. Elements of $U \times V$ with non-zero membership grade in $R(U, V)$ are represented in the diagram by lines connecting the respective nodes. Although not explicitly shown, the lines have membership values equal to unity.

Example 2.10 (Mendel 1995a) Let R represent the relation of *stability* between the set of all linear, second-order continuous-time systems and the set of the poles of such systems. Of all the possible pairings of linear second-order continuous-time systems and poles, only those pairs whose members are time-invariant with poles lying either in the left-half of the complex s-plane or on the imaginary axis of that plane are known to be stable.

Let $U = \{u_1, u_2\}$ = {linear second-order time-varying continuous-time system, linear second-order time-invariant continuous-time system}, and $V = \{y_1, y_2, y_3\}$ = {poles lie in the left-half s-plane, poles lie on the $j\omega$ axis, poles lie in the right-half s-plane}. The Cartesian product $U \times V$ can be visualized as a 2×3 array of ordered pairs, e.g., the (1, 2) element is (linear second-order time-varying continuous-time system, poles lie on the $j\omega$ axis). The stability relation $R(U, V)$ is the following subset of $U \times V$:

$$\begin{aligned} R(U,V) = \{&(\text{linear second-order time-invariant continuous-time system, poles lie in the left-half } s\text{-plane}),\\ &(\text{linear second-order time-invariant continuous-time system, poles lie on the } j\omega \text{ axis} \end{aligned}$$

The relational matrix for this stability relation is:

$$\begin{array}{cc} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} u_1 \\ u_2 \end{array} & \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$$

The sagittal diagram for this stability relation is depicted in Fig. 2.10.

Let $R(u, v)$ and $S(u, v)$—R and S for short—be two crisp relations in the same Cartesian product space $U \times V$. The intersection and union of R and S, which are *compositions* of the two relations, are computed using (2.17) and (2.18), because a relation is a set.

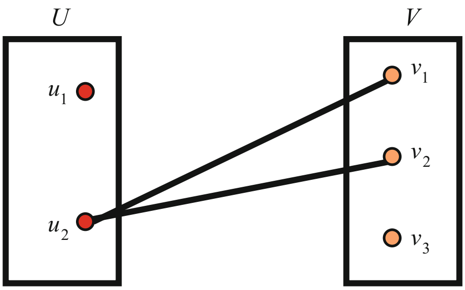


Fig. 2.10 Sagittal diagram for relation of stability between the set of all linear, second-order continuous-time systems and the set of poles of such systems (Mendel 1995a © 1995, IEEE)

2.6 Fuzzy Relations and Compositions on the Same Product Space

A fuzzy relation represents a degree *of presence or absence of association, interaction, or interconnectedness between the elements of two or more fuzzy sets.* Some examples of binary fuzzy relations are:

- x is much larger than y
- y is very close to x
- z is much greener than y
- system 1 is less damped than system 2
- bandwidth of system A is larger than that of system B
- tone C is of higher local signal-to-noise ratio than tone D
- a is more profitable than b.

A binary type-1 fuzzy relation $F(A_1, A_2)$ is (Lin and Lee 1996) a type-1 fuzzy set that is defined on the Cartesian product space of crisp sets A_1 and A_2, where tuples (a_1, a_2) may have varying degrees of membership $\mu_F(a_1, a_2)$ within the relation. More specifically,

$$F(A_1, A_2) = \int_{A_1 \times A_2} \mu_F(a_1, a_2)/(a_1, a_2) \quad a_1 \in A_1 \text{ and } a_2 \in A_2 \tag{2.30}$$

where $\mu_F(a_1, a_2) \in [0, 1]$. It is important to note that the elements of $F(A_1, A_2)$ are numbers and not fuzzy sets. When they are type-1 fuzzy sets, the fuzzy relation becomes a type-2 fuzzy relation (Sect. 7.6).

Example 2.11 (Mendel 1995a) Let U and V be the real numbers, and consider the fuzzy relation "target x is *close* to target y." Here is one MF for this relation:

$$\mu_c(|x - y|) \equiv \max\{(5 - |x - y|)/5, 0\} \tag{2.31}$$

This relational MF is depicted in Fig. 2.11. Note that the distance between the two targets $|x - y|$ is treated as the independent variable.

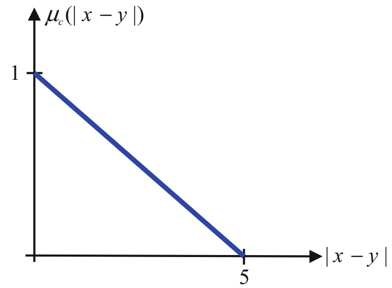


Fig. 2.11 Relational MF $\mu_c(|x - y|)$ (Mendel 1995a © 1995, IEEE)

Because fuzzy relations are fuzzy sets in a Cartesian product space, set theoretic and algebraic operations can be defined for them using the earlier operators for fuzzy union, intersection and complement. Let $R(U, V)$ and $S(U, V)$ (shortened in the sequel to R and S) be two fuzzy relations in the *same* Cartesian product space $U \times V$. The intersection and union of R and S, which are *compositions* of the two relations, are then defined as:

$$\mu_{R\cap S}(x, y) = \mu_R(x, y) \bigstar \mu_S(x, y) \tag{2.32}$$

$$\mu_{R\cup S}(x, y) = \mu_R(x, y) \oplus \mu_S(x, y) \tag{2.33}$$

where $\bigstar$ is any t-norm, and $\oplus$ is any t-conorm.

Example 2.12 Consider the two somewhat contradictory fuzzy relations "u is close to v" *and* "u is smaller than v," and also the less-contradictory relations "u is close to v" *or* "u is smaller than v." All relations are on the same Cartesian product space $U \times V$. For simplicity, it is assumed here that $U = \{u_1, u_2\} = \{2, 12\}$ and $V = \{v_1, v_2, v_3\} = \{1, 7, 13\}$. Let the MFs for *close* and *smaller* be denoted as $\mu_c(u, v)$ and $\mu_s(u, v)$, respectively, where the numbers in $\mu_c(u, v)$ and $\mu_s(u, v)$ have been chosen to agree with a comparison of the numbers in U and V.

$$\mu_c(u, v) \equiv \begin{array}{c c} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} u_1 \\ u_2 \end{array} & \begin{pmatrix} 0.9 & 0.4 & 0.1 \\ 0.1 & 0.4 & 0.9 \end{pmatrix} \end{array} \tag{2.34}$$

and

$$\mu_s(u, v) \equiv \begin{array}{c c} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} u_1 \\ u_2 \end{array} & \begin{pmatrix} 0 & 0.6 & 1 \\ 0 & 0 & 0.3 \end{pmatrix} \end{array} \tag{2.35}$$

The membership grades for the union and intersection of these relations, assuming minimum t-norm ($\wedge$) and maximum t-conorm ($\vee$), can be found as (i = 1, 2 and j = 1, 2, 3)

$$\mu_{c\cup s}(u_i, v_j) = \mu_c(u_i, v_j) \vee \mu_s(u_i, v_j) \tag{2.36}$$

and

$$\mu_{c\cap s}(u_i, v_j) = \mu_c(u_i, v_j) \wedge \mu_s(u_i, v_j) \tag{2.37}$$

Using (2.36) and (2.37), it is easy to show that

$$\mu_{c\cup s}(u,v) = \begin{array}{c} \\ u_1 \\ u_2 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{pmatrix} 0.9 & 0.6 & 1 \\ 0.1 & 0.4 & 0.9 \end{pmatrix} \end{array} \tag{2.38}$$

and

$$\mu_{c\cap s}(u,v) = \begin{array}{c} \\ u_1 \\ u_2 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{pmatrix} 0 & 0.4 & 0.1 \\ 0 & 0 & 0.3 \end{pmatrix} \end{array} \tag{2.39}$$

From (2.38) and (2.39), "u is close to v" *or* "u is smaller than v" is seen to be much more sensible than "u is close to v" *and* "u is smaller than v," because membership values in $\mu_{c\cup s}(u,v)$ are fairly large, whereas those in $\mu_{c\cap s}(u,v)$ are mostly small.

2.7 Crisp Relations and Compositions on Different Product Spaces

Consider[14] two different product spaces, $U \times V$ and $V \times W$, that share a common set and let $R(U,V)$ and $S(V,W)$ be two *crisp* relations on these spaces. The composition of these relations is defined (Klir and Folger 1988, p. 75) as: *a subset $T(U,W)$ of $U \times W$ such that $(u,w) \in T$ if and only if $(u,v) \in R$ and $(v,w) \in S$.* This can be expressed as a *max–min composition*, *max–product composition* or, in general, as the following *sup-star composition* for crisp relations:

$$\mu_{R\circ S}(u,w) = \sup_{v\in V}[\mu_R(u,v)\bigstar\mu_S(v,w)]\; u \in U, w \in W \tag{2.40}$$

where $\star$ indicates any suitable t-norm operation. The validity of the *sup-star* composition for crisp set is shown in Wang (1997, p. 54). If R and S are two crisp relations on $U \times W$ and $V \times W$, respectively, then the membership for any pair (u,w), $u \in U$ and $w \in W$, is 1 if and only if there exists at least one $v \in V$ such that $\mu_R(u,v) = 1$ and $\mu_S(v,w) = 1$. In Zadeh (1973) it is shown that this condition is equivalent to having the sup-star composition equal to 1. Because this is a special case of the sup-star composition for fuzzy sets (a crisp set is a special case of a fuzzy set), whose proof is given in Sect. 2.8, the proof of (2.40) for crisp sets is not included here.

[14] Most of this paragraph is taken from Karnik and Mendel (2001, p. 337).

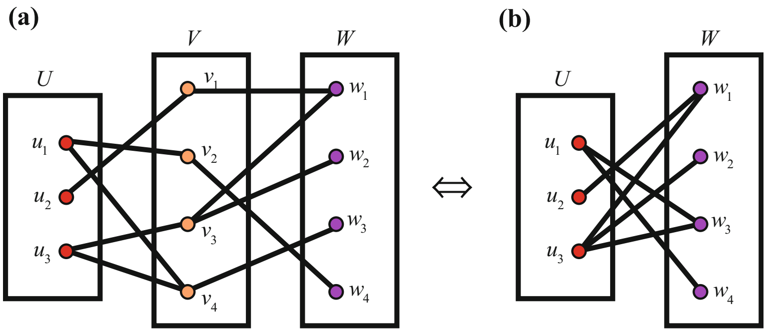


Fig. 2.12 Saggital diagram for Example 2.13. **a** Original diagram for relations $R_1(U, V)$ and $R_2(V, W)$ and **b** compositional diagram for $R_3(U, W)$ (Mendel 1995a © 1995, IEEE)

Example 2.13 Given the saggital diagrams depicted in Fig. 2.12a, b, one concludes that the relational matrices $R_1(U, V)$, $R_2(V, W)$, and $R_3(U, W)$ are:

$$R_1(U,V) = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{array} \tag{2.41}$$

$$R_2(V,W) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{2.42}$$

$$R_3(U,W) = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \end{array} \begin{array}{c} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array} \tag{2.43}$$

Because it is not efficient to keep describing compositions in terms of sagittal diagrams, a *formula* is needed that conveys the same information.

Definition 2.8 *The* max–min composition *of relations* $R(U, V)$ *and* $S(V, W)$ *is defined by the MF* $\mu_{R \circ S}(u, w)$*, where*

$$\mu_{R\circ S}(u,w) = \left\{ (u,w), \max_v [\min(\mu_R(u,v), \mu_S(v,w))] \right\} \tag{2.44}$$

The max-product composition *of relations* $R(U, V)$ *and* $S(V, W)$ *is defined by the MF* $\mu_{R\times S}(u, w)$, *where*

$$\mu_{R\times S}(u, w) = \left\{(u, w), \max_{v}[\mu_R(u, v)\mu_S(v, w)]\right\} \tag{2.45}$$

Clearly, the *max–min* or *max–product* compositions lead to the correct relational matrix $R(U, W)$, because they are special cases of the sup-star composition in (2.40).

Example 2.14 Here (2.44) and (2.45) are verified for the (1, 2) element of $R_3(U, W)$ in (2.43). For this element, (2.44) becomes

$$\begin{aligned}\mu_{R_3}(u_1, w_2) &= \left\{(u_1, w_2), \max_{v}[\min(\mu_{R_1}(u_1, v), \mu_{R_2}(v, w_2))]\right\}\\ &= \{(u_1, w_2), \max[\min(\mu_{R_1}(u_1, v_1), \mu_{R_2}(v_1, w_2)),\\ &\quad \min(\mu_{R_1}(u_1, v_2), \mu_{R_2}(v_2, w_2)), \min(\mu_{R_1}(u_1, v_3), \mu_{R_2}(v_3, w_2)),\\ &\quad \min(\mu_{R_1}(u_1, v_4), \mu_{R_2}(v_4, w_2))]\}\\ &= \{(u_1, w_2), \max[\min(0, 0), \min(1, 0), \min(0, 1), \min(1, 0)]\}\\ &= \{(u_1, w_2), \max[0, 0, 0, 0]\} = \{(u_1, w_2), 0\}\end{aligned} \tag{2.46}$$

which agrees with (2.43). Similarly, (2.45) becomes

$$\begin{aligned}\mu_{R_3}(u_1, w_2) &= \left\{(u_1, w_2), \max_{v}[\mu_{R_1}(u_1, v)\mu_{R_2}(v, w_2)]\right\}\\ &= \{(u_1, w_2), \max[\mu_{R_1}(u_1, v_1)\mu_{R_2}(v_1, w_2), \mu_{R_1}(u_1, v_2)\mu_{R_2}(v_2, w_2),\\ &\quad \mu_{R_1}(u_1, v_3)\mu_{R_2}(v_3, w_2), \mu_{R_1}(u_1, v_4)\mu_{R_2}(v_4, w_2)]\}\\ &= \{(u_1, w_2), \max[(0\times 0), (1\times 0), (0\times 1), (1\times 0)]\}\\ &= \{(u_1, w_2), \max[0, 0, 0, 0]\} = \{(u_1, w_2), 0\}\end{aligned} \tag{2.47}$$

which also agrees with (2.43).

The following shortcuts can be used to evaluate the *max-min* or *max-product* compositions that involve relational matrices.

- **Max–min composition**: (1) Write out each element in the matrix product $Q(U, V)P(V, W)$; but, (2) treat each multiplication as a minimum operation; and, then, (3) treat each addition as a maximum operation.
- **Max–product composition**: (1) Write out each element in the *matrix product* $Q(U, V)P(V, W)$; but, (2) treat each multiplication as an algebraic multiplication operation; and, then, (3) treat each addition as a maximum operation.

Example 2.14 (Continued) Here these two shortcuts are used to again verify (2.44) and (2.45), but this time for the (1, 3) element of $R_3(U, W)$ in (2.43). Now applying the shortcut for the max-min composition to the $(1, 3)$ element of $R_1(u, v) \times R_2(v, w)$, one finds

$$\begin{aligned} R_3(u_1, w_3) &= 0 \times 0 + 1 \times 0 + 0 \times 0 + 1 \times 1 \\ &= \min(0, 0) + \min(1, 0) + \min(0, 0) + \min(1, 1) \\ &= \max(0, 0, 0, 1) = 1 \end{aligned} \tag{2.48}$$

Similarly, applying the shortcut for the max-product composition to the (1, 3) element of $R_1(u, v) \times R_2(v, w)$, one finds

$$\begin{aligned} R_3(u_1, w_3) &= 0 \times 0 + 1 \times 0 + 0 \times 0 + 1 \times 1 \\ &= \max(0, 0, 0, 1) = 1 \end{aligned} \tag{2.49}$$

Both of these results agree with the (1, 3) element of $R_3(U, W)$ in (2.43).

The *max–min* and *max–product* compositions are not the only ones that correctly represent $R(U, W)$; however, they seem to be the most widely used ones.

2.8 Fuzzy Relations and Compositions on Different Product Spaces

Next, consider the composition of fuzzy relations from different Cartesian product spaces that share a common set, namely $R(U, V)$ and $S(V, W)$, e.g., u is *smaller* than v, and v is *close* to w. The composition of fuzzy relations from different Cartesian product spaces that share a common set is defined analogously to the crisp composition, except that in the fuzzy case the sets are fuzzy sets. Associated with fuzzy relation R is its MF $\mu_R(u, v)$, where $\mu_R(u, v) \in [0, 1]$; and, associated with fuzzy relation S is its MF $\mu_S(v, w)$, where $\mu_S(v, w) \in [0, 1]$. In this respect, the condition on the composition of crisp relations, that is given below (2.40), can be rephrased as follows:

Theorem 2.1 *If*[15] *R and S are two type-1 fuzzy relations on $U \times V$ and $V \times W$, respectively, then the membership for any pair (u, w), $u \in U$ and $w \in W$, is non-zero if and only if there exists at least one $v \in V$ such that $\mu_R(u, v) \neq 0$ and $\mu_S(v, w) \neq 0$, i.e.:*

$$\mu_{R \circ S}(u, w) = \sup_{v \in V}[\mu_R(u, v) \bigstar \mu_S(v, w)]\ u \in U, v \in V \tag{2.50}$$

(2.50) *is called the* sup-star composition *for fuzzy relations*.

Proof This proof uses the following method: Let **A** be the statement "$\mu_{R \circ S}(u, w) \neq 0$," and **B** be the statement "there exists at least one $v \in V$ such that

[15]This theorem and its proof are taken from Karnik and Mendel, (1998, pp. 61–62).

$\mu_R(u,v) \neq 0$ *and* $\mu_S(v,w) \neq 0$." "**A** if and only if **B**" is proved by first proving that[16] $\bar{\mathbf{B}} \Rightarrow \bar{\mathbf{A}}$ (equivalent to proving that $\mathbf{A} \Rightarrow \mathbf{B}$, i.e., necessity of **B**) and then proving that $\bar{\mathbf{A}} \Rightarrow \bar{\mathbf{B}}$ (equivalent to proving that $\mathbf{B} \Rightarrow \mathbf{A}$, i.e., sufficiency of **B**).

Necessity—If there exists no $v \in V$ such that $\mu_R(u,v) \neq 0$ *and* $\mu_S(v,w) \neq 0$, then this means that for every $v \in V$, either $\mu_R(u,v)$ or $\mu_S(v,w)$ is equal to zero (or both are zero), which in turn implies that $\mu_R(u,v)\bigstar\mu_S(v,w) = 0$ for every $v \in V$; hence, the supremum[17] of $\mu_R(u,v)\bigstar\mu_S(v,w)$ over $v \in V$ is also zero, and therefore $\mu_{R\circ S}(u,w) = 0$, which is $\bar{\mathbf{A}}$.

Sufficiency—If the sup-star composition is zero then it must be true that $\mu_R(u,v)\bigstar\mu_S(v,w) = 0$ for every $v \in V$, which means that for every $v \in V$, either $\mu_R(u,v)$ or $\mu_S(v,w)$ (or both) is zero. This means that there is no $v \in V$ such that $\mu_R(u,v) \neq 0$ and $\mu_S(v,w) \neq 0$, which is $\bar{\mathbf{B}}$.

When R and S are from discrete universes of discourse, then $R \circ S$ can be described either by a sagittal diagram, in which each branch is labeled by its MF value, or a fuzzy relational matrix, in which each element is a positive real number between and including zero and unity. When U, V, and W are discrete universes of discourse, then the supremum operation in (2.50) is the *maximum*. Although it is permissible to use other t-norms, the most commonly used sup-star compositions are the *sup-min* and *sup-product*. The shortcuts for computing the sup-min and sup-product, given in Sect. 2.7, apply also to fuzzy compositions over discrete universes of discourse.

Example 2.15 Consider the type-1 relation "u is close to v" on $U \times V$, where $U = \{u_1, u_2\}$ and $V = \{v_1, v_2, v_3\}$ are given in Example 2.12 as U = {2, 12} and V = {1, 7, 13}, and $\mu_c(u,v)$ is given by (2.34). Now consider another type-1 fuzzy relation "v is much bigger than w" on $V \times W$, where $W = \{w_1, w_2\} = \{4, 8\}$, with the following MF, $\mu_{mb}(v,w)$, for *much bigger*, where the numbers in $\mu_{mb}(v,w)$ have been chosen to agree with a comparison of the numbers in V and W:

$$\mu_{mb}(v,w) \equiv \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{cc} w_1 & w_2 \end{array} \\ \begin{pmatrix} 0 & 0 \\ 0.6 & 0 \\ 1 & 0.7 \end{pmatrix} \end{array} \tag{2.51}$$

The statement "u is close to v *and* v is much bigger than w" indicates the composition of these two type-1 relations. This composition can be found by using (2.50) and, e.g., the minimum t-norm, as follows ($i = 1, 2$ and $j = 1, 2, 3$):

[16]Recall that $\overline{\text{at least } (\cdot)} = \text{no } (\cdot)$.

[17]Let S be a set of real numbers. An upper bound for S is a number b such that $x \leq b$ for all $x \in S$. The *supremum* of S, if it exists, is the smallest upper bound for S. An upper bound that actually belongs to the set is called *a maximum*.

$$\begin{aligned}\mu_{comb}(u_i, w_j) = & \left[\mu_c(u_i, v_1) \wedge \mu_{mb}(v_1, w_j)\right] \vee \left[\mu_c(u_i, v_2) \wedge \mu_{mb}(v_2, w_j)\right] \\ & \vee \left[\mu_c(u_i, v_3) \wedge \mu_{mb}(v_3, w_j)\right]\end{aligned} \tag{2.52}$$

where $\wedge$ denotes minimum, and $\vee$ denotes maximum. For example,

$$\begin{aligned}\mu_{comb}(u_1, w_1) &= [\mu_c(u_1, v_1) \wedge \mu_{mb}(v_1, w_1)] \vee [\mu_c(u_1, v_2) \wedge \mu_{mb}(v_2, w_1)] \\ &\quad \vee [\mu_c(u_1, v_3) \wedge \mu_{mb}(v_3, w_1)] \\ &= [0.9 \wedge 0] \vee [0.4 \wedge 0.6] \vee [0.1 \wedge 1] \\ &= 0 \vee 0.4 \vee 0.1 = 0.4\end{aligned} \tag{2.53}$$

Doing all the calculations in a similar manner, one finds (Exercise 2.17):

$$\mu_{comb}(u, w) = \begin{array}{c} \\ u_1 \\ u_2 \end{array} \begin{array}{c} \begin{array}{cc} w_1 & w_2 \end{array} \\ \begin{pmatrix} 0.4 & 0.1 \\ 0.9 & 0.7 \end{pmatrix} \end{array} \tag{2.54}$$

Unlike the case of crisp compositions, for which exactly the same results are obtained using either the max-min or max-product compositions, the same results are not obtained in the case of fuzzy compositions. This is a major difference between fuzzy and crisp compositions.

Suppose fuzzy relation R is just a fuzzy set, in which case $V = U$, so that $\mu_R(u, v)$ just becomes $\mu_R(u)$ [or $\mu_R(v)$], e.g., "v is *medium large* and v is *smaller than* w." What happens to the sup-star composition in this case? Because $V = U$,

$$\sup_{v \in V}[\mu_R(u, v) \bigstar \mu_S(v, w)] = \sup_{u \in U}[\mu_R(u) \bigstar \mu_S(u, w)] \tag{2.55}$$

which is only a function of output variable w; hence, the notation $\mu_{R \circ S}(u, w)$ can be simplified to $\mu_{R \circ S}(w)$, so that *when R is just a fuzzy set*,

$$\mu_{R \circ S}(w) = \sup_{u \in U}[\mu_R(u) \bigstar \mu_S(u, w)] \quad w \in W \tag{2.56}$$

Eq. (2.56), which is also known as Zadeh's *compositional rule of inference* (Zadeh 1973), is used a lot in Chap. 3 as the type-1 inference mechanism for a rule.

Example 2.16 Consider again the Example 2.12 relation "u is close to v" on $U \times V$, where $U = \{2, 12\}$ and $V = \{1, 7, 13\}$. The MF for $\mu_c(u, v)$ is given in (2.34). Let the fuzzy set "small" on U be defined as

$$\mu_s(u) \equiv \begin{array}{c} \begin{array}{cc} u_1 & u_2 \end{array} \\ \begin{pmatrix} 0.9 & 0.1 \end{pmatrix} \end{array} \tag{2.57}$$

The composition of the two statements "u is small *and* u is close to v" can be obtained by using (2.56) as follows ($j = 1, 2, 3$):

$$\mu_{s\circ c}(v_j) = [\mu_s(u_1) \wedge \mu_c(u_1, v_j)] \vee [\mu_s(u_2) \wedge \mu_c(u_2, v_j)] \tag{2.58}$$

Using (2.57), it is straightforward to show that

$$\begin{array}{cccc} & v_1 & v_2 & v_3 \\ \mu_{s\circ c}(v) = & (\,0.9 & 0.4 & 0.1\,) \end{array} \tag{2.59}$$

For discrete universes of discourse, the max–min or max-product compositions in (2.56) can be evaluated using the shortcuts described earlier; however, first a row matrix for $\mu_R(u)$ must be created i.e., if $u \in U = \{u_1, u_2, \ldots, u_n\}$ and $R(U) = (\mu_R(u_1), \mu_R(u_2), \ldots, \mu_R(u_n))$, then:

- **Max–min composition**: (1) Write out each element in the matrix product $R(U)S(U, W)$, but (2) treat each multiplication as a minimum operation, and then (3) treat each addition as a maximum operation.
- **Max-product composition**: (1) Write out each element in the *matrix product* $R(U)S(U, W)$, but (2) treat each multiplication as an algebraic multiplication operation, and then (3) treat each addition as a maximum operation.

2.9 Hedges

A *linguistic hedge* or modifier,[18] introduced first in Zadeh (1972), is an operation that modifies the meaning of a term, or more generally, of a fuzzy set. For example, if *weak pressure* is a fuzzy set, then *very weak pressure*, *more-or-less weak pressure*, *extremely weak pressure*, and *not-so weak pressure* are examples of hedges that are applied to this fuzzy set. There are a multitude of hedges, many additional examples of which can be found in Schmucker (1984), Cox (1994).

There are two ways to handle a hedge:

1. They can be viewed as operators that act on a fuzzy set's MF to modify it.
2. They can be treated as new linguistic terms.

By the first approach, one establishes a set of primary terms and their MFs. The hedge operators then operate on some[19] or all of the primary terms, leading to a larger set of terms and their MFs. Three hedge operators introduced in Zadeh (1972) are:

1. *Concentration*: $\mu_{con(F)}(x) \equiv [\mu_F(x)]^2$. If, e.g., *weak pressure* has MF $\mu_{WP}(p)$, then *very weak pressure* is a fuzzy set with MF $[\mu_{WP}(p)]^2$, and *very very weak pressure* is a fuzzy set with MF $[\mu_{WP}(p)]^4$. Because MFs have been assumed to

[18] Some of the material in this section is taken from Mendel and Wu (2010, Sect. 3.6) and Mendel (1995a, p. 356).

[19] Hedges should only operate on primary terms for which the hedged term makes linguistic sense, e.g. the hedge *much* makes no linguistic sense when it is applied to the primary term *low pressure*.

be normalized, it is clear that the operation of concentration leads to a MF that lies within the MF of the original fuzzy set (thus, the term *concentration*); both have the same support, and the same membership values where the value of the original MF equals unity or zero.

2. *Dilation*: $\mu_{dil(F)}(x) \equiv [\mu_F(x)]^{1/2}$. If, e.g., *weak pressure* has MF $\mu_{WP}(p)$, then *more or less weak pressure* is a fuzzy set with MF $[\mu_{WP}(p)]^{1/2}$. The operation of dilation leads to a MF that lies outside of the MF of the original fuzzy set (thus, the term *dilation*); both have the same support, and the same membership values where the value of the original MF equals unity or zero.
3. *Artificial Hedges*: Two hedges that are quite useful are the *plus* and *minus* hedges, whose MFs are $\mu_{plus(F)}(x) \equiv [\mu_F(x)]^{1.25}$ and $\mu_{minus(F)}(x) \equiv [\mu_F(x)]^{0.75}$. These artificial hedges provide milder degrees of concentration and dilation than those associated with the concentration and dilation hedges.

The $\equiv$ sign has been used in these hedge MF formulas to convey the fact that their exponents are quite arbitrary; they can be changed depending upon one's interpretation of the hedges,[20] as already noticed in Zadeh (1972), who stated:

> It should be emphasized, however, that these representations are intended mainly to illustrate the approach rather than to provide accurate definitions of the hedges in question. Furthermore it must be understood that our analysis and its conclusions are tentative in nature and may require modification in later work.

The following example illustrates the use of hedge operators.

Example 2.17 (Adapted from Zadeh 1973, p. 35) In conversations, one frequently uses the phrase *highly unlikely*. Here it is shown how to obtain a MF for it. Let X denote a universe of discourse associated with an appropriate quantity related to the notion of *likely*. X is clarified below. Let $\mu_{LIKELY}(x)$ be the MF for the term *likely*. Then,

$$\mu_{HIGHLY-UNLIKELY}(x) = [1 - \mu_{LIKELY}(x)]^{4\times 0.75} \tag{2.60}$$

To obtain (2.60), the hedge *highly* has been interpreted as *minus very very* (which, of course, is subjective) and the fact that *unlikely* is the complement of *likely* has also been used.

From estimation theory (e.g., Edwards (1972); Mendel (1995b)), it is known that *likelihood is proportional to probability*. This fact helps us to establish the universe of discourse, X, as values of probability (the constant of proportionality between probability and likelihood is irrelevant), i.e., $x \in X = [0, 1]$. As a concrete example, assume the following discrete universe of discourse: $X = \{0, 0.1, 0.2, 0.3, \cdots, 0.9, 1\}$. To evaluate (2.60), $\mu_{LIKELY}(x)$ needs to be specified. Based on my perception of the fuzzy set *likely*, the following ad hoc choice is made for $\mu_{LIKELY}(x)$ (your choice may be different):

[20]Because of the uncertainty about the numerical values of the exponents, hedges might be more appropriately modeled within the framework of type-2 fuzzy sets. This is examined in Sect. 7.10.

$$\mu_{LIKELY}(x) \equiv 1/1 + 1/0.9 + 1/0.8 + 0.8/0.7 + 0.6/0.6 + 0.5/0.5 + 0.3/0.4 + 0.2/0.3 \tag{2.61}$$

Recall that the terms not shown have zero MF values. Evaluating (2.60), one finds that

$$\mu_{HIGHLY-UNLIKELY}(x) \approx 1/0 + 1/0.1 + 1/0.2 + 0.5/0.3 + 0.3/0.4 \tag{2.62}$$

Observe, from (2.61) and (2.62), that the MF $\mu_{HIGHLY-UNLIKELY}(x)$ seems to make sense, i.e., it agrees with the notion that something highly unlikely has a very very small chance (i.e., probability) of occurring. Consequently, large values for $\mu_{HIGHLY-UNLIKELY}(x)$ should and indeed do occur for small values of probability, x.

In Schmucker (1984) one finds the following:

> Representing hedges as operators acting upon the representation of the primary terms has both positive and negative implications. On the positive side, it seems very natural and also allows for an easy implementation of the connection of several hedges…. The negative side of representing hedges as operators is that some hedges don't seem to be easily modeled by such an approach. By this we mean that the way people normally use these hedges entails an implementation considerably different and more complex than that of an operator that acts uniformly upon the fuzzy restrictions that represent the various primary terms.

Finally, in Macvicar-Whelen (1978) there are experimental results that indicate the hedge *very* causes a *shift* in the MF rather than a steepening of the MF as is obtained by the concentration operator; hence, their paper calls into question the use of operators to model hedges.

In the rest of this book, unless otherwise indicated, hedges are treated as new linguistic terms.

2.10 Extension Principle

The Extension Principle was introduced in[21] Zadeh (1975) and is an important tool in fuzzy set theory.[22] Heavy use is made of it in later chapters of this book. It extends mathematical relationships between non-fuzzy variables to fuzzy variables. Suppose, for example, that the MF for the fuzzy set *small* is given and the MF for the fuzzy set $(small)^2$ is desired. The Extension Principle determines the MF for

[21]According to Klir and Yuan (1995), the Extension Principle was introduced in Zadeh (1975); however, Zadeh (1975, p. 236, footnote 18) states that the Extension Principle is implicit in a result given in Zadeh (1965).

[22]Actually, there are other Extension Principles (e.g., He, et al. 2000; Arabi, et al. 2001), but the one that is described in this section is the most widely used and is the one used in the rest of this book.

$(small)^2$ by making use of the non-fuzzy mathematical relationship $y = x^2$, in which the fuzzy set *small* plays the role of x, and also the MF for *small*.

Suppose one is given a function of a single variable x, $y = f(x)$, where $x \in U$ and $y \in V$. For illustrative purposes, U is assumed to be a discrete universe of discourse, U_d, and

$$A = \sum_{x \in U_d} \mu_A(x)/x \tag{2.63}$$

The Extension Principle states that (Jang et al. 1997) the image of the fuzzy set A under the mapping $f(\cdot)$ can be expressed as a fuzzy set B, i.e.,

$$\begin{aligned} B = f(A) = f\left(\sum_{x \in U_d} \mu_A(x)/x\right) \\ = \mu_A(x_1)/y_1 + \mu_A(x_2)/y_2 + \cdots + \mu_A(x_N)/y_N \equiv \mu_B(y) \end{aligned} \tag{2.64}$$

where $(i = 1, \ldots, N)$ $y_i = f(x_i)$. Since $x = f^{-1}(y)$, where $f^{-1}(y)$ is the inverse of f (i.e., $f[f^{-1}(y)] = y$), another way to express B is by $\mu_B(y) = \mu_A[f^{-1}(y)]$, $y \in V$.

Example 2.18 As a concrete illustration of (2.64), suppose that U_d = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, and $A = small = 1/1 + 0.8/2 + 0.6/3 + 0.3/4$; then, $B = (small)^2 = 1/1 + 0.8/4 + 0.6/9 + 0.3/16$.

The version of the Extension Principle given in (2.64) is valid only if the mapping between y and $f(x)$ is one-to-one. It is quite possible that the same value of y can be obtained for different values of x—a many-to-one mapping—in which case (2.64) needs to be modified, e.g., $f(x_1) = f(x_2) = y$, but $x_1 \neq x_2$ and $\mu_A(x_1) \neq \mu_A(x_2)$. To resolve this ambiguity, the larger one of the two membership values is assigned to $\mu_B(y)$. The general modification to (2.64) is Wang (1997):

$$\mu_B(y) = \max_{x \in f^{-1}(y)} \mu_A(x) \quad y \in V \tag{2.65}$$

where $f^{-1}(y)$ denotes the set of all points $x \in U$ such that $f(x) = y$.

Example 2.19 As an illustration of (2.65), suppose that $U_d = \{-3, -2, -1, 0, 1, 2\}$ and fuzzy set A is characterized by the MF values listed in the second column of Table 2.1. Then $\mu_B(y)$, for $y = f(x) = x^4$, is given in the last column of that table.

Table 2.1 Numerical results for Example 2.19

x	$\mu_A(x)$	$y = f(x) = x^4$	$\mu_B(y)$
−3	0.5	81	$\max\{0.5\} = 0.5$
−2	0.6	16	$\max\{0.6, 0.1\} = 0.6$
−1	1.0	1	$\max\{1, 0.4\} = 1$
0	0.9	0	$\max\{0.9\} = 0.9$
1	0.4	1	$\max\{1, 0.4\} = 1$
2	0.1	16	$\max\{0.6, 0.1\} = 0.6$

Observe that there are two pairs of elements of U that map into the same value of y: -2 and 2 map into 16, and -1 and 1 map into 1. In both cases the membership value of y is obtained by taking the maximum of the membership grades of the respective two elements. From the last two columns of Table 2.1, observe that $B = 0.9/0 + 1/1 + 0.6/16 + 0.5/81$.

So far the Extension Principle has been stated just for a mapping of a single variable. Things get a bit more complicated for a function of more than one variable. Suppose, for example, one has a function of two variables x_1 and x_2, i.e., $y = f(x_1, x_2)$, where $x_1 \in X_{d1}$, $x_2 \in X_{d2}$, $y \in V$, X_{d1} and X_{d2} are assumed to be discrete universes of discourse, and:

$$A_1 = \sum_{x_1 \in X_{d1}} \mu_{A_1}(x_1)/x_1 \tag{2.66}$$

and

$$A_2 = \sum_{x_2 \in X_{d2}} \mu_{A_2}(x_2)/x_2 \tag{2.67}$$

Now it is possible for $y = f(x_1, x_2)$ to be many-to-one, just as it was in the single-variable case; so, the Extension Principle for the two-variable case needs to look something like (2.65). The difference between the two- and one-variable cases is that in the latter there is only one MF that can be evaluated for each value of x, whereas in the former there are two MFs that can be evaluated, namely $\mu_{A_1}(x_1)$ and $\mu_{A_2}(x_2)$. In this case, the Extension Principle becomes:

$$\mu_{f(A_1,A_2)}(y) \equiv \mu_B(y) = \begin{cases} \sup\limits_{(x_1,x_2)\in f^{-1}(y)} & \min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2)\} \\ 0 & \text{if } f^{-1}(y) = \emptyset \end{cases} \tag{2.68}$$

where $f^{-1}(y)$ now denotes the set of all points $x_1 \in X_{d1}$ and $x_2 \in X_{d2}$ such that $f(x_1, x_2) = y$. The condition in (2.68) that $\mu_B(y) = 0$ if $f^{-1}(y) = \emptyset$ means that if there are no values of x_1 and x_2 for which a specific value of y can be reached, then the MF value for that specific value of y is set equal to zero. Only those values of y that satisfy $y = f(x_1, x_2)$ can be reached. Note that (Yager 1986) provides a justification of (2.68) based on the sup-star composition.[23]

Example 2.20 [Adapted from Lin and Lee (1996, p. 30)] As an illustration of (2.68), suppose that $X_{d1} = \{-1, 0, 1\}$ and $X_{d2} = \{-2, 2\}$, and fuzzy sets A_1 and A_2

[23]A *plausibility argument* for the Extension Principle is: (1) $y = f(x_1, x_2)$ can be interpreted literally, as: When $x_1 = x_1'$ *and* $x_2 = x_2'$ then $y = f(x_1', x_2')$, where the *and* in this statement is modeled as a conjunction, which explains the use of the minimum in (2.68); and, (2) when $y = f(x_1, x_2)$ is many-to-one, then this can be interpreted as: For $(x_1, x_2) = (x_1^1, x_2^1)$ *or* (x_1^2, x_2^2) *or* … *or* (x_1^m, x_2^m), the same value is obtained for $y = f(x_1, x_2)$, where the *or*'s in this statement are modeled as disjunctions, which explains the use of the maximum (sup) in (2.68).

Table 2.2 Numerical results for Example 2.20

x_1	$\mu_{A_1}(x_1)$	x_2	$\mu_{A_2}(x_2)$	$y = f(x_1, x_2) = x_1^2 + x_2$	$\mu_{f(A_1,A_2)}(y) \equiv \mu_B(y)$
−1	0.5	−2	0.4	−1	$\max\{0.4, 0.4\} = 0.4$
−1	0.5	2	1.0	3	$\max\{0.5, 0.9\} = 0.9$
0	0.1	−2	0.4	−2	$\max\{0.1\} = 0.1$
0	0.1	2	1.0	2	$\max\{0.1\} = 0.1$
1	0.9	−2	0.4	−1	$\max\{0.4, 0.4\} = 0.4$
1	0.9	2	1.0	3	$\max\{0.5, 0.9\} = 0.9$

are characterized by the MFs listed in the second and fourth columns of Table 2.2. Then the MF for the fuzzy set B that is associated with $\mu_{f(A_1,A_2)}(y)$, where $y = f(x_1, x_2) = x_1^2 + x_2$, is given in the last column of that table. The construction of this table first required determining all x_1 and x_2 pairs for which y is defined. These values constitute the Cartesian product of X_{d1} and X_{d2}, $X_{d1} \times X_{d2}$. By evaluating $y = f(x_1, x_2) = x_1^2 + x_2$ at all these values, it is established that $V = \{-2, -1, 2, 3\}$.

There are two ordered pairs (−1, −2) and (1, −2) that map into the same value of y, namely −1, and, there are also two ordered pairs (−1, 2) and (1, 2) that map into the same value of y = 3. It is for these two sets of ordered pairs that the respective maximum membership grades must be taken in (2.68).

The calculations of $\mu_B(y)$ are illustrated next for $y = -1$:

$$\begin{aligned}\mu_B(-1) &= \max[\min\{\mu_{A_1}(-1), \mu_{A_2}(-2)\}, \min\{\mu_{A_1}(1), \mu_{A_2}(-2)\} \\ &= \max[\min(0.5, 0.4), \min(0.9, 0.4)] = 0.4\end{aligned} \tag{2.69}$$

From the last two columns of Table 2.2, one concludes that $B = 0.1/-2 + 0.4/-1 + 0.1/2 + 0.9/3$.

Finally, the generalization of the Extension Principle in (2.68) from 2 to r variables is considered. The Cartesian product of r arbitrary non-fuzzy sets $X_1, X_2, \ldots, X_r$, denoted by $X_1 \times X_2 \times \cdots \times X_r$, is the non-fuzzy set of all ordered r-tuples $(x_1, x_2, \ldots, x_r)$ such that $x_i \in X_i$ for $i \in \{1, 2, \ldots, r\}$; i.e., Rudin (1966)

$$X_1 \times \cdots \times X_r = \{(x_1, \ldots, x_r) | x_1 \in X_1, \ldots, x_r \in X_r\}$$

Let f be a mapping from $X_1 \times \cdots \times X_r$ to a universe Y such that $y = f(x_1, \ldots, x_r) \in Y$, and $A_1, A_2, \ldots, A_r$ be type-1 fuzzy sets in $X_1, X_2, \ldots, X_r$, respectively. Then, Zadeh's Extension Principle allows one to induce from the r type-1 fuzzy sets $A_1, A_2, \ldots, A_r$ a type-1 fuzzy set B on Y, through f, i.e., $B = f(A_1, A_2, \ldots, A_r)$, such that (see 2.68)[24]

[24]Equation (2.70) assumes that $x_1, \ldots, x_r$ are *non-interactive* (e.g., if $x_1 = a$ and $x_2 = a^2$, then x_1 and x_2 are interactive) or that there is no joint constraint on $x_1, \ldots, x_r$. For a detailed discussion about this, see Zadeh (1975), Appendix B in Karnik and Mendel (1998) and Rajati and Mendel (2013).

$$\mu_B(y) = \begin{cases} \sup\limits_{(x_1,\ldots,x_r)\in f^{-1}(y)} & \min\{\mu_{A_1}(x_1),\ldots,\mu_{A_r}(x_r)\} \\ 0 & \text{if } f^{-1}(y) = \emptyset \end{cases} \tag{2.70}$$

where $f^{-1}(y)$ denotes the set of all points $x_1 \in X_1,\ldots,x_r \in X_r$ such that $y = f(x_1,\ldots,x_r)$.

To implement (2.70), first the values of $x_1,\ldots,x_r$ must be found for which $y = f(x_1,\ldots,x_r)$, after which $\mu_{A_1}(x_1),\ldots,\mu_{A_r}(x_r)$ and $\min\{\mu_{A_1}(x_1),\ldots,\mu_{A_r}(x_r)\}$ are computed at those values. If more than one set of $x_1,\ldots,x_r$ satisfy $y = f(x_1,\ldots,x_r)$, then this procedure is repeated for all of them and the largest of the minima is chosen as the choice for $\mu_B(y)$.

Zadeh defined the Extension Principle using minimum t-norm and maximum t-conorm (for the supremum operation). Other t-norms and t-conorms can be used, as described, e.g., in Dubois and Prade (1980). In this book, only the maximum t-conorm and either the minimum or product t-norms are used. Note that when the minimum in (2.70) is replaced by another t-norm, the sup-min composition is replaced by the sup-star composition.

When one needs to extend an operation of the form $f(x_1,\ldots,x_r)$ to an operation $f(A_1,\ldots,A_r)$ (e.g., $A_1 + \cdots + A_r$) where A_i are type-1 fuzzy sets, the individual operations like multiplication, addition, etc., involved in *f*, are not extended. Instead, the following definition is used, which derives directly from (2.70) when the maximum operation is used for the union and a general t-norm (★) is used instead of the minimum operation:

$$f(A_1,\ldots,A_r) = \int\limits_{x_1\in X_1} \cdots \int\limits_{x_r\in X_r} \mu_{A_1}(x_1)\bigstar\cdots\bigstar\mu_{A_r}(x_r)/f(x_1,\ldots,x_r) \tag{2.71}$$

For example, if $f(x_1,x_2) = x_1x_2/(x_1+x_2)$, the extension of *f* to type-1 fuzzy sets A_1 and A_2 is written as:

$$f(A_1,A_2) = \int\limits_{x_1\in X_1}\int\limits_{x_2\in X_2} \mu_{A_1}(x_1)\bigstar\mu_{A_2}(x_2)\bigg/\frac{x_1x_2}{x_1+x_2} \tag{2.72}$$

and **not** as $f(A_1,A_2) = A_1 \times A_2/(A_1+A_2)$.

To compute $f(A_1,\ldots,A_r)$ using (2.71), $f(x_1,\ldots,x_r)$ and $\mu_{A_1}(x_1)\bigstar\cdots\bigstar\mu_{A_r}(x_r)$ must be computed for $\forall x_1 \in X_1,\ldots,\forall x_r \in X_r$. It is easy to write a computer program to do this, although sometimes it can be done analytically, as is demonstrated in the next three examples.

Example 2.21 Let F_1, …, and F_n be type-1 interval fuzzy numbers having domains $[l_1,r_1]$, …, and $[l_n,r_n]$, respectively. Then $\sum_{i=1}^n F_i$ is also a type-1 interval fuzzy number whose domain is $\left[\sum_{i=1}^n l_i, \sum_{i=1}^n r_i\right]$. The proof is by mathematical induction.

(a) For F_1 and F_2, using (2.71), the algebraic sum of F_1 and F_2 can be obtained as

$$F_1 + F_2 = \int_{u \in [l_1, r_1]} \int_{w \in [l_2, r_2]} (1 \bigstar 1)/(u + w) \tag{2.73}$$

Observe from (2.73) that: (1) each term in $F_1 + F_2$ is equal to the sum $u + w$ for some $u \in [l_1, r_1]$ and $w \in [l_2, r_2]$, the smallest term being $(l_1 + l_2)$ and the largest being $(r_1 + r_2)$; and (2) since both F_1 and F_2 have continuous domains, $F_1 + F_2$ has a continuous domain; hence, $F_1 + F_2$ is a type-1 interval fuzzy number with domain $[l_1 + l_2, r_1 + r_2]$.

(b) The proof of the general result is straightforward, and is left to the reader (Exercise 2.26).

Example 2.22 Let $F_1, \ldots,$ and F_n be type-1 interval fuzzy numbers having domains $[l_1, r_1], \ldots,$ and $[l_n, r_n]$, respectively. Then, $\sum_{i=1}^n a_i F_i + b$ (where each a_i as well as b is a positive real number) is also a type-1 interval fuzzy number whose domain is $\left[\sum_{i=1}^n a_i l_i + b, \sum_{i=1}^n a_i r_i + b\right]$. The derivation of these results follows.

Consider $F_i = 1/[l_i, r_i]$. Multiplying F_i by the positive real number a$_i$ (expressed as the type-1 fuzzy set $1/a_i$) yields [use (2.71)][25]

$$a_i F_i = \int_{v \in V} 1/(a_i v) \;\; V = [l_i, r_i] \tag{2.74}$$

Adding the positive real number b (expressed as the type-1 fuzzy set $1/b$) to $a_i F_i$ yields (see 2.73)

$$a_i F_i + b = \int_{v \in V} 1/(a_i v + b) \;\; V = [l_i, r_i] \tag{2.75}$$

Substituting $w = a_i v + b$ into (2.75), it follows that:

$$a_i F_i + b = \int_{w \in W} 1/w \;\; W = [a_i l_i + b, a_i r_i + b] \tag{2.76}$$

Consequently, from Example 2.21 and (2.76), the domain of $\sum_{i=1}^n a_i F_i + b$ is $\left[\sum_{i=1}^n a_i l_i + b, \sum_{i=1}^n a_i r_i + b\right]$, Q. E. D.

Note that, when $[l_i, r_i]$ is expressed in terms of its center and spread, as $[c_i - s_i, c_i + s_i]$, for which $l_i = c_i - s_i$ and $r_i = c_i + s_i$, then $\left[\sum_{i=1}^n a_i l_i + b, \sum_{i=1}^n a_i r_i + b\right] = \left[\sum_{i=1}^n a_i c_i + b - \sum_{i=1}^n a_i s_i, \sum_{i=1}^n a_i c_i + b + \sum_{i=1}^n a_i s_i\right]$, which is sometimes a useful alternate way to express the domain of $\sum_{i=1}^n a_i F_i + b$.

[25]Note that $1 \bigstar 1 = 1$ regardless of whether the t-norm is minimum or product.

Exercise 2.27 asks the reader to obtain the comparable results when a_i are positive or negative real numbers.

Example 2.23 Given n type-1 Gaussian fuzzy sets $F_1, \ldots, F_n$, with means $m_1, \ldots, m_n$ and standard deviations $\sigma_1, \ldots, \sigma_n$, i.e.,

$$F_i = \int_{x \in X} \exp\left[-\frac{1}{2}\left(\frac{x - m_i}{\sigma_i}\right)^2\right] \bigg/ x \quad i = 1, \ldots, n \tag{2.77}$$

The affine combination $\sum_{i=1}^{n} a_i F_i + b$, where a_i and b are crisp constants, is also a type-1 Gaussian fuzzy set with mean $\sum_{i=1}^{n} a_i m_i + b$ and standard deviation Σ', where

$$\Sigma' = \begin{cases} \sqrt{\sum_{i=1}^{n} a_i^2 \sigma_i^2} & \text{if product t-norm is used} \\ \sum_{i=1}^{n} |a_i \sigma_i| & \text{if minimum t-norm is used} \end{cases} \tag{2.78}$$

The proofs of these results, which can be found in Karnik and Mendel (1998, Appendix C.9), use the results from Exercises 2.24 and 2.25.

2.11 α-Cuts[26]

In the first edition of this book there was no material about α-cuts, because both type-1 and interval type-2 rule-based systems did not need them. Beginning in Sect. 6.7.3, it will be seen that α-cuts play a central role for general type-2 fuzzy sets and systems, something that was not known when the first edition of this book was written.

Definition 2.9 (Zadeh 1975) The *α-cut of type-1 fuzzy set A*, denoted A_α, is an interval of real numbers, defined as:

$$A_\alpha = \{x | \mu_A(x) \geq \alpha\} \tag{2.79}$$

where $\alpha \in [0, 1]$.

Example 2.24 An example of an *α-cut* is depicted in Fig. 2.13, and in this example, $A_\alpha = [1.9, 5.5]$. Observe that the α-cut lies on the x-axis.

Example 2.25 Given a specific type-1 fuzzy set A, it is easy to obtain formulas for the end-points of an α-cut, e.g. see Table 2.3. In order to obtain these formulas, such as the ones for the triangular distribution, solve the two equations $l(x) = \alpha$ for the left end-point and $r(x) = \alpha$ for the right end-point of A_α.

[26]If a reader is interested only in type-1 and interval type-2 fuzzy sets and systems, this section, as well as Sects. 2.12 and 2.13, can be omitted.

Fig. 2.13 A trapezoidal type-1 fuzzy set and an α-cut (Mendel and Wu 2010 © 2010, IEEE)

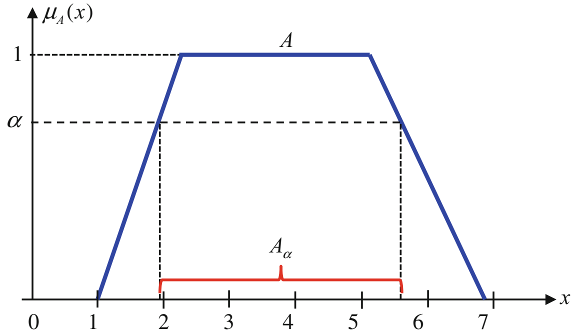
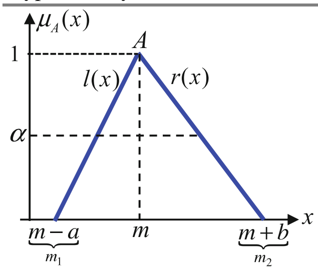
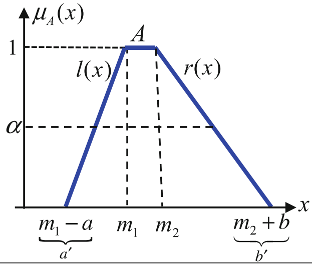


Table 2.3 Examples of type-1 fuzzy sets and their α-cut formulas (Mendel and Wu 2010 © 2010, IEEE)

Type-1 fuzzy set	$\alpha-$cut formula
	$A_\alpha = [a_\alpha, b_\alpha] = [m - a(1-\alpha), m + b(1-\alpha)] = [m_1 + (m - m_1)\alpha, m_2 - (m_2 - m)\alpha]$
	$A_\alpha = [a_\alpha, b_\alpha] = [m_1 - a(1-\alpha), m_2 + b(1-\alpha)] = [a' + (m_1 - a')\alpha, b' - (b' - m_2)\alpha]$

Theorem 2.2 *The following set-theoretic properties hold for* α*-cuts*:

$$(A \cap B)_\alpha = A_\alpha \cap B_\alpha \tag{2.80}$$

$$(A \cup B)_\alpha = A_\alpha \cup B_\alpha \tag{2.81}$$

Equations (2.80) and (2.81) state that the α-cut of the intersection (union) of two type-1 fuzzy sets equals the intersection (union) of their α-cuts.

Proof Because the proof of (2.81) is so similar to the proof of (2.80), only the proof (2.80) is provided here; the proof of (2.81) is left as an exercise (Exercise 2.28). This proof is taken from Klir and Yuan (1995, p. 35), and is given for the minimum intersection operator.

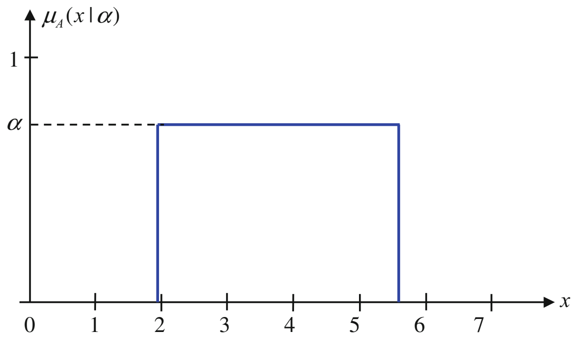


Fig. 2.14 Square-well function $\mu_A(x|\alpha)$ (Mendel and Wu 2010 © 2010, IEEE)

For any $x \in (A \cap B)_\alpha$, it follows from Definition 2.9 that $\mu_{A \cap B}(x) \geq \alpha$; hence, $\min[\mu_A(x), \mu_B(x)] \geq \alpha$. This means $\mu_A(x) \geq \alpha$ and $\mu_B(x) \geq \alpha$ which implies $x \in A_\alpha \cap B_\alpha$, and consequently $(A \cap B)_\alpha \subseteq A_\alpha \cap B_\alpha$.

Conversely, for any $x \in A_\alpha \cap B_\alpha$, $x \in A_\alpha$ and $x \in B_\alpha$. This means, again from Definition 2.9, that $\mu_A(x) \geq \alpha$ and $\mu_B(x) \geq \alpha$; hence, $\min[\mu_A(x), \mu_B(x)] \geq \alpha$ which means $\mu_{A \cap B}(x) \geq \alpha$. This implies $x \in (A \cap B)_\alpha$, and consequently $A_\alpha \cap B_\alpha \subseteq (A \cap B)_\alpha$.

Combining the two parts of this proof, one concludes that $(A \cap B)_\alpha = A_\alpha \cap B_\alpha$.

2.12 Representing Type-1 Fuzzy Sets Using α-Cuts

One of the major roles of α-cuts is their capability to represent a type-1 fuzzy set. In order to do this, first the following *indicator function* is introduced:

$$I_{A_\alpha}(x) = \begin{cases} 1 & x \in A_\alpha \\ 0 & x \notin A_\alpha \end{cases} \tag{2.82}$$

Associated with $I_{A_\alpha}(x)$ is the following *square-well function*:

$$\mu_A(x|\alpha) \equiv \alpha I_{A_\alpha}(x) = \alpha / A_\alpha \tag{2.83}$$

This function, an example of which is depicted in Fig. 2.14, raises the α-cut A_α off of the x-axis to height (level) α.

Theorem 2.3 (Decomposition Theorem) *A type-1 fuzzy set A can be represented as:*

$$\mu_A(x) = \bigcup_{\alpha \in [0,1]} \mu_A(x|\alpha) = \sup_{\alpha \in [0,1]} \{\alpha / A_\alpha\} \; x \in X \tag{2.84}$$

where $\cup$ *is the fuzzy union (i.e., sup over* $[0, 1]$*).*

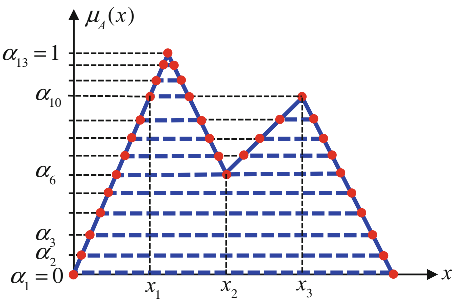


Fig. 2.15 Example to illustrate the Decomposition Theorem when 13 α-cuts are used

This theorem was introduced in Zadeh (1971) and also appears in Zadeh (1975, p. 223), where it is called a *resolution identity*. It is also called a "Decomposition Theorem" because A is decomposed into a collection of square-well functions (i.e., intervals raised to level α) that are then aggregated using the union operation (with respect to α). An example of (2.84) is depicted in Fig. 2.15. In that figure: (1) the blue dashed lines are the α-cuts raised to level α; (2) the red dots show $\mu_A(x)$ computed by using (2.84); and, (3) at x_1, x_2 and x_3, the dashed vertical lines intersect many of the dashed blue lines, but they terminate at their maximum values, the respective red dot, according to (2.84).

Theorem 2.3 holds for continuous and discrete universes of discourse, since (2.84) is valid for both universes, and is valid for convex and non-convex type-1 fuzzy sets. Note that greater resolution is obtained by including more α-cuts, and the calculation of new α-cuts does not affect previously calculated α-cuts. A proof of Theorem 2.3 can be found, e.g., in Klir and Yuan (1995, p. 41) or Wang (1997, p. 369). It is not included herein because, once one understands (2.84), it becomes a rather obvious result.

Example 2.26 (Taken from Mendel, et al. 2014, p. 38) Let $A = 0.2/x_1 + 0.4/x_2 + 0.6/x_3 + 0.8/x_4 + 1/x_5$. Some indicator functions for A are:

$$\begin{aligned}
I_{A_{0.2}}(x) &= 1/x_1 + 1/x_2 + 1/x_3 + 1/x_4 + 1/x_5 \\
I_{A_{0.4}}(x) &= 0/x_1 + 1/x_2 + 1/x_3 + 1/x_4 + 1/x_5 \\
I_{A_{0.6}}(x) &= 0/x_1 + 0/x_2 + 1/x_3 + 1/x_4 + 1/x_5 \\
I_{A_{0.8}}(x) &= 0/x_1 + 0/x_2 + 0/x_3 + 1/x_4 + 1/x_5 \\
I_{A_{1.0}}(x) &= 0/x_1 + 0/x_2 + 0/x_3 + 0/x_4 + 1/x_5
\end{aligned} \tag{2.85}$$

Their associated square-well functions are:

$$\begin{aligned}
\mu_A(x|0.2) &= 0.2/x_1 + 0.2/x_2 + 0.2/x_3 + 0.2/x_4 + 0.2/x_5 \\
\mu_A(x|0.4) &= 0/x_1 + 0.4/x_2 + 0.4/x_3 + 0.4/x_4 + 0.4/x_5 \\
\mu_A(x|0.6) &= 0/x_1 + 0/x_2 + 0.6/x_3 + 0.6/x_4 + 0.6/x_5 \\
\mu_A(x|0.8) &= 0/x_1 + 0/x_2 + 0/x_3 + 0.8/x_4 + 0.8/x_5 \\
\mu_A(x|1.0) &= 0/x_1 + 0/x_2 + 0/x_3 + 0/x_4 + 1/x_5
\end{aligned} \tag{2.86}$$

Applying (2.84) to these functions, it follows that:

$$A = \mu_A(x|0.2) \cup \mu_A(x|0.4) \cup \mu_A(x|0.6) \cup \mu_A(x|0.8) \cup \mu_A(x|1.0) \tag{2.87}$$

When performing these unions, focus on a specific domain point, e.g. $x = x_4$, for which

$$\mu_A(x_4) = \max(0, 0.2, 0.4, 0.6, 0.8)/x_4 = 0.8/x_4 \tag{2.88}$$

Performing these unions for the five domain points, whose MFs are non-zero, it is straightforward to recover $A = 0.2/x_1 + 0.4/x_2 + 0.6/x_3 + 0.8/x_4 + 1/x_5$.

Example 2.27 For a *convex type-1 fuzzy set*, such as the ones in Table 2.3, $A_\alpha = [a_\alpha, b_\alpha]$ $(\alpha \in [0, 1])$, and (2.84) can be expressed as:

$$\mu_A(x) = \sup_{\alpha \in [0,1]} \{\alpha/[a_\alpha, b_\alpha]\} \; x \in X \tag{2.89}$$

The following is a corollary to Theorems 2.2 and 2.3:

Corollary 2.1 *The intersection and union of type-1 fuzzy sets A and B can be computed by using their α-cuts, as follows:*

$$\mu_{A \cap B}(x) = \bigcup_{\alpha \in [0,1]} \alpha/(A_\alpha \cap B_\alpha) \tag{2.90}$$

$$\mu_{A \cup B}(x) = \bigcup_{\alpha \in [0,1]} \alpha/(A_\alpha \cup B_\alpha) \tag{2.91}$$

Proof From Theorem 2.3, it follows that:

$$\mu_{A \cap B}(x) = \bigcup_{\alpha \in [0,1]} \mu_{A \cap B}(x|\alpha) = \bigcup_{\alpha \in [0,1]} \alpha/(A \cap B)_\alpha \tag{2.92}$$

Applying (2.80) to (2.92), it follows that:

$$\mu_{A \cap B}(x) = \bigcup_{\alpha \in [0,1]} \alpha/(A_\alpha \cap B_\alpha) \tag{2.93}$$

which is (2.90). Because the proof of (2.91) is so similar to the proof of (2.90) it is not provided here.

Equation (2.90) is also true when $\cap$ is replaced by t-norm symbol $\bigstar$. An important feature of (2.90) and (2.91) is that, since A_α and B_α are intervals (or multiple intervals) of real numbers, $A_\alpha \cup B_\alpha$ and $A_\alpha \cap B_\alpha$ are easily computed.

Example 2.28 Here (2.91) is applied to the two type-1 fuzzy sets A and B that are depicted in Fig. 2.16a, to verify that the correct answer is obtained for $\mu_{A \cup B}(x)$. The union of A and B, computed as $\max(\mu_A(x), \mu_B(x))$ is depicted as the red curve

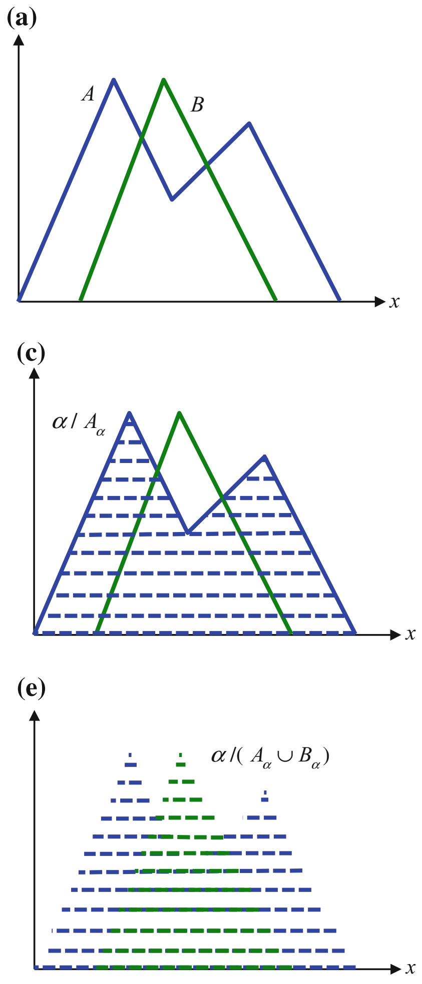


Fig. 2.16 **a** Type-1 fuzzy sets A and B, **b** $A \cup B = \max\{\mu_A(x), \mu_B(x)\}$, **c** α-cuts of A raised to level α **d** α-cuts of B raised to level α, **e** $A_\alpha \cup B_\alpha$ raised to level α, and **f** $A \cup B = \bigcup_{\alpha \in [0,1]} \alpha/(A_\alpha \cup B_\alpha)$

in Fig. 2.16b. Some α-cuts that are raised to level α for A and B, are depicted (as the dashed lines) in Fig. 2.16c, d, respectively. By superimposing all of these dashed lines for α/A_α and α/B_α one obtains[27] $\alpha/(A_\alpha \cup B_\alpha)$ in Fig. 2.16e. The envelope of all of the $\alpha/(A_\alpha \cup B_\alpha)$ in Fig. 2.16e provides the red curve in Fig. 2.16f, which is in agreement with the red curve in Fig. 2.16b. Each point on the red envelope can be obtained by going to a specific value of x, drawing a

[27]Recall that $A_\alpha \cup B_\alpha$ is a set of real numbers that includes all elements in either A_α or B_α.

vertical line up from it, and choosing the height of that line as the value of the highest dashed horizontal line that intersects it.

2.13 Functions of Type-1 Fuzzy Sets Computed by Using α-Cuts

Recall[28] (Sect. 2.10) that the Extension Principle states that when the function $y = f(x_1, x_2, \ldots, x_r)$ is applied to type-1 fuzzy sets X_i $(i = 1, \ldots, r)$ the result is another type-1 fuzzy set, Y, whose MF $\mu_Y(y)$ is given by (2.70). Because $\mu_Y(y)$ is a type-1 fuzzy set, it can, therefore, be expressed in terms of its α-cuts as follows (see (2.79), (2.82)–(2.84), where Y_α plays the role of A_α):

$$Y_\alpha = \{y | \mu_Y(y) \geq \alpha\} \tag{2.94}$$

$$I_{Y_\alpha}(x) = \begin{cases} 1 & y \in Y_\alpha \\ 0 & y \notin Y_\alpha \end{cases} \tag{2.95}$$

$$\mu_Y(y|\alpha) \equiv \alpha I_{Y_\alpha}(y) = \alpha / Y_\alpha \tag{2.96}$$

$$\mu_Y(y) = \bigcup_{\alpha \in [0,1]} \mu_Y(y|\alpha) = \sup_{\alpha \in [0,1]} \{\alpha / Y_\alpha\} \quad y \in D_Y \tag{2.97}$$

In order to implement (2.95)–(2.97), a method is needed to compute Y_α, and this is provided in the following:

Theorem 2.4 (α-Cuts Decomposition Theorem[29]) *Let $Y = f(X_1, X_2, \ldots, X_r)$ be an arbitrary (crisp) function, where X_i $(i = 1, \ldots, r)$ is a type-1 fuzzy set whose domain is D_{X_i} and α-cut is $(X_i)_\alpha$. Then, under the Extension Principle*:

$$Y_\alpha = f\big((X_1)_\alpha, \ldots, (X_2)_\alpha\big) \tag{2.98}$$

and the height of Y equals the minimum height of all X_i.

Equation (2.98) shows that the α-cut of a function of type-1 fuzzy sets equals that function applied to the α-cuts of those type-1 fuzzy sets. Theorem 2.4 does not address how to compute $f\big((X_1)_\alpha, \ldots, (X_2)_\alpha\big)$. Example 2.30 below shows how to do this for a specific nonlinear function, and, when this theorem is used in later chapters of this book for other functions, explanations will be given for how to

---

[28] Much of the material in this section (up to Example 2.29) is taken from Mendel and Wu (2010, Sect. 5A.2, © IEEE 2010).

[29] The statement of this theorem is adapted from Klir and Yuan (1995, Theorem 2.9) and is taken from Mendel and Wu (2010). Zadeh (1975) states this result without a proof for it. Nguyen (1978) seems to be the first to provide necessary and sufficient conditions for (2.98) to hold.

compute $f((X_1)_\alpha, \ldots, (X_2)_\alpha)$ for those functions. It is no exaggeration to say that this theorem is now vitally important for general type-2 fuzzy systems.

Proof For all $y \in D_Y$, from (2.94) it follows that[30]

$$y \in Y_\alpha \Leftrightarrow \mu_Y(y) \geq \alpha \tag{2.99}$$

Under the Extension Principle in (2.70),

$$\mu_Y(y) \geq \alpha \Leftrightarrow \sup_{(x_1,\ldots,x_r)|y=f(x_1,\ldots,x_r)} \min\{\mu_{X_1}(x_1), \ldots, \mu_{X_r}(x_r)\} \geq \alpha \tag{2.100}$$

It follows that:

$$\begin{aligned}
&\sup_{(x_1,\ldots,x_r)|y=f(x_1,\ldots,x_r)} \min\{\mu_{X_1}(x_1), \ldots, \mu_{X_r}(x_r)\} \geq \alpha \\
&\Leftrightarrow (\exists x_{10} \in D_{X_1} \text{ and } \cdots \text{ and } x_{r0} \in D_{X_r}) \text{ such that} \\
&\quad (y = f(x_{10}, \ldots, x_{r0}) \text{ and } \min\{\mu_{X_1}(x_{10}), \ldots, \mu_{X_r}(x_{r0})\} \geq \alpha) \\
&\Leftrightarrow (\exists x_{10} \in D_{X_1} \text{ and } \cdots \text{ and } x_{r0} \in D_{X_r}) \text{ such that} \\
&\quad (y = f(x_{10}, \ldots, x_{r0}) \text{ and } [\mu_{X_1}(x_{10}) \geq \alpha \text{ and } \cdots \text{ and } \mu_{X_r}(x_{r0}) \geq \alpha]) \\
&\Leftrightarrow (\exists x_{10} \in D_{X_1} \text{ and } \cdots \text{ and } x_{r0} \in D_{X_r}) \text{ such that} \\
&\quad \left(y = f(x_{10}, \ldots, x_{r0}) \text{ and } \left[x_{10} \in (X_1)_\alpha \text{ and } \cdots \text{ and } x_{r0} \in (X_r)_\alpha\right]\right) \\
&\Leftrightarrow y \in f((X_1)_\alpha, \ldots, (X_2)_\alpha)
\end{aligned} \tag{2.101}$$

Hence, from the last line of (2.101) and (2.100),

$$\mu_Y(y) \geq \alpha \Leftrightarrow y \in f((X_1)_\alpha, \ldots, (X_2)_\alpha) \tag{2.102}$$

which is (2.98). Because the right-hand side of (2.100) (read from right to the left) indicates that α cannot exceed the minimum height of all $\mu_{X_i}(x_i)$ (otherwise there is no α-cut on one or more X_i), the height of Y must equal the minimum height of all X_i.

Example 2.29 Let[31] $A = [a, b, c]$ and $B = [p, q, r]$ be two triangle type-1 fuzzy numbers whose MFs are:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \end{cases} \tag{2.103}$$

[30]This proof is similar to the one that is given for Theorem 2.9 in Klir and Yuan (1995), where it is only provided for a function of a single variable. Even so, our proof of Theorem 2.4 follows the proof of their Theorem 2.9 very closely; however, their theorem does not explain how sub-normal type-1 fuzzy sets should be handled. Such sub-normal type-1 fuzzy sets are quite common in type-2 fuzzy sets because many kinds of lower MFs (see Chap. 6) are sub-normal.

[31]This example is adapted from Dutta, et al. (2011).

$$\mu_B(x) = \begin{cases} \frac{x-p}{q-p} & p \le x \le q \\ \frac{r-x}{r-q} & q \le x \le r \end{cases} \tag{2.104}$$

Then the α-cuts of A and B are (use the first row of Table 2.3 in which $m_1 = a$, $m = b$ and $m_2 = c$):

$$A_\alpha = [(b-a)\alpha + a, c - (c-b)\alpha] \tag{2.105}$$

$$B_\alpha = [(q-p)\alpha + p, r - (r-q)\alpha] \tag{2.106}$$

Here the MF of $A+B$, the sum of two type-1 fuzzy numbers, is computed. To begin, the $\alpha -$ cuts of A and B are added using interval arithmetic, namely

$$[r+s] + [t+u] = [r+t, s+u] \tag{2.107}$$

Consequently:

$$\begin{aligned} A_\alpha + B_\alpha &= [(b-a)\alpha + a, c-(c-b)\alpha] + [(q-p)\alpha + p, r-(r-q)\alpha] \\ &= [a+p+(b-a+q-p)\alpha, c+r-(c-b+r-q)\alpha] \end{aligned} \tag{2.108}$$

To find $\mu_{A+B}(x)$ equate to x both the first and second components in (2.108) [note that this is the reverse of what was done to obtain the α-cuts in (2.105) and (2.106)]:

$$x = a + p + (b - a + q - p)\alpha \tag{2.109a}$$

$$x = c + r - (c - b + r - q)\alpha \tag{2.109b}$$

Next, express α in terms of x and then set $\alpha = 0$ and $\alpha = 1$ in (2.109a, 2.109b) to obtain a respective value of α together with the respective domain of x, as:

$$\alpha = \frac{x-(a+p)}{(b+q)-(a+p)}, \quad (a+p) \le x \le (b+q) \tag{2.110a}$$

$$\alpha = \frac{(c+r)-x}{(c+r)-(b+q)}, \quad (b+q) \le x \le (c+r) \tag{2.110b}$$

Because α is the MF grade of $A+B$ (this is a crucial observation) it follows that:

$$\mu_{A+B}(x) = \begin{cases} \frac{x-(a+p)}{(b+q)-(a+p)} & (a+p) \le x \le (b+q) \\ \frac{(c+r)-x}{(c+r)-(b+q)} & (b+q) \le x \le (c+r) \end{cases} \tag{2.111}$$

Observe that $A+B$ is also a type-1 fuzzy number, i.e. $A+B = [(a+p), (b+q), (c+r)]$.

A very interesting exposition about interval computing (e.g., using α - cuts) and fuzzy sets is Kreinovich (2008).

2.14 Multivariable MFs and Cartesian Products

Most discussions in this chapter have been for type-1 fuzzy sets that depend on only one variable. This section describes how to characterize type-1 fuzzy sets that depend on up to p variables, $x_1, x_2, \ldots, x_p$.

For two variables, x_1 and x_2, type-1 fuzzy set A is defined on the Cartesian product $X_1 \times X_2$, i.e.,

$$\begin{aligned} A &= \{((x_1, x_2), \mu_A(x_1, x_2)) | x_1 \in X_1, x_2 \in X_2\} \\ &= \{((x_1, x_2), \mu_A(x_1, x_2)) | (x_1, x_2) \in X_1 \times X_2\} \end{aligned} \tag{2.112}$$

where $\mu_A(x_1, x_2)$ is a general function of x_1 and x_2. When $X_1 \times X_2$ is continuous, then A can also be written as

$$A = \int_{x_1 \in X_1} \int_{x_2 \in X_2} \mu_A(x_1, x_2)/(x_1, x_2) \tag{2.113}$$

or, if $X_1 \times X_2$ is discrete, $X_{1d} \times X_{2d}$, then A can be written as

$$A = \sum\nolimits_{x_1 \in X_{1d}} \sum\nolimits_{x_2 \in X_{2d}} \mu_A(x_1, x_2)/(x_1, x_2) \tag{2.114}$$

When the MF $\mu_A(x_1, x_2)$ is *separable*, which occurs when x_1 and x_2 do not interact with one another, then it is expressed in terms of $\mu_{A_1}(x_1)$ and $\mu_{A_2}(x_2)$, as

$$\mu_A(x_1, x_2) = \mu_{A_1}(x_1) \bigstar \mu_{A_2}(x_2) \tag{2.115}$$

where $\star$ denotes a t-norm such as minimum or product. In this book only separable MFs are used.

The extensions of these two-variable results to more than two variables is straightforward, e.g., for p variables, when the MF $\mu_A(x_1, x_2, \ldots, x_p)$ is separable, then

$$\mu_A(x_1, x_2, \ldots, x_p) = \mu_{A_1}(x_1) \bigstar \mu_{A_2}(x_2) \bigstar \cdots \bigstar \mu_{A_p}(x_p) \tag{2.116}$$

where $x_1 \in X_1, x_2 \in X_2, \ldots, x_p \in X_p$, which can be interpreted as the Cartesian product of the type-1 fuzzy sets $A_1, A_2, \ldots, A_p$ in the product space $X_1 \times X_2 \times \cdots \times X_p$.

Equation (2.116) is frequently written as

$$\mu_A(x_1, x_2, \ldots, x_p) = \mu_{X_1}(x_1)\bigstar\mu_{X_2}(x_2)\bigstar\cdots\bigstar\mu_{X_p}(x_p) \tag{2.117}$$

Using the notation of (2.117), X_i plays a double role as the label of the fuzzy set and as the universe of discourse for x_i. Usually, this does not cause any confusion. (2.117) is widely used in Chap. 3.

2.15 Crisp Logic

According to the *Encyclopedia Britannica*, "Logic is the study of propositions and their use in argumentation." According to *Webster's Dictionary of the English Language*, "logic is the science of formal reasoning, using principles of valid inference," and " logic is the science whose chief end is to ascertain the principles on which all valid reasoning depends, and which may be applied to test the legitimacy of every conclusion that is drawn from premises." Although multi-valued logic exists, most of us are most familiar with two-valued (dual-valued) logic in which a proposition is either *true* or *false*. With the advent of fuzzy logic, this kind of logic is also referred to as *crisp logic*, which was first systematized by Aristotle thousands of years ago, in ancient Athens.

From Fig. 1.2, observe that one of the major components of a fuzzy system is *Rules*. In this book, rules will be expressed as logical implications, i.e., in the forms of IF–THEN statements, e.g.,

$$\text{IF } x \text{ is } A, \text{THEN } y \text{ is } B, \text{where } x \in X \text{ and } y \in Y$$

A rule represents a special kind of *relation* between A and B; its MF is denoted $\mu_{A\rightarrow B}(x, y)$. What is a proper and appropriate choice for this MF? Nothing that has been presented so far helps us to answer this question, because an implication resides within a branch of mathematics known as logic, and so far only set theory has been discussed. Fortunately, as stated in Klir and Folger (1988, p. 24):

> It is well established that propositional logic is isomorphic to set theory under the appropriate correspondence between components of these two mathematical systems. Furthermore, both of these systems are isomorphic to a Boolean algebra, which is a mathematical system defined by abstract (interpretation-free) entities and their axiomatic properties. … The isomorphisms between Boolean algebra, set theory, and propositional logic guarantee that every theorem in any one of these theories has a counterpart in each of the other two theories. … These isomorphisms allow us, in effect, to cover all these theories by developing only one of them.

Consequently, not a lot of time will be spent reviewing crisp logic; but, some time must be spent on it, especially on the concept of implication, in order to reach the comparable concept in fuzzy logic.

Rules are a form of propositions.[32] A *proposition* is an ordinary statement involving terms that have been defined, e.g., "The damping ratio is low." Consequently, one could have the following rule: "IF the damping ratio is low, THEN the system's impulse response oscillates a long time before it dies out." In traditional propositional logic, a proposition must be meaningful to call it "true" or "false," whether or not one knows which of these terms properly applies.

Logical reasoning is the process of combining given propositions into other propositions, and then doing this over and over again. Propositions can be combined in many ways, all of which are derived from three fundamental operations: *conjunction* (denoted $p \wedge q$), where one asserts the simultaneous truth of two separate propositions p and q (e.g., damping ratio is low and bandwidth is large); *disjunction* (denoted $p \vee q$), where one asserts the truth of either or both of two separate propositions (e.g., I will design an analog filter or I will design a digital filter); and, *implication* (denoted $p \rightarrow q$), which usually takes the form of an IF–THEN rule, an example of which has been given in the previous paragraph. The IF part of an implication is called its *antecedent* whereas the THEN part is called its *consequent*.

In addition to generating propositions using conjunction, disjunction, or implication, a new proposition can be obtained from a given one by prefixing the clause "it is false that …". This is the operation of *negation* (denoted $\sim p$). Additionally, $p \leftrightarrow q$ is the *equivalence* relation; it means that p and q are both true or false.

In traditional propositional logic an implication is said to be *true* if one of the following holds: (1) antecedent is true, consequent is true, (2) antecedent is false, consequent is false, and (3) antecedent is false, consequent is true. The implication is called *false* when (4) antecedent is true, consequent is false. Situation (1) is the familiar one of common experience. Situation (2) is also reasonable, for if one starts from a false assumption one expects to reach a false conclusion, however, intuition is not always reliable. One may reason correctly from a false antecedent to a true consequent (e.g., IF 1 = 2, THEN 3 = 3; note that 1 = 2 is false, but, adding 2 = 1 to this false statement, lets one correctly conclude that 3 = 3); hence, a false antecedent can lead to a consequent which is either true or false, and thus both situations (2) and (3) are allowed in traditional propositional logic. Finally, situation (4) is in accord with our intuition, for an implication is clearly false if a true antecedent leads to a false consequent.

A logical structure is constructed by applying the aforementioned five operations to propositions. The objective of a logical structure is to determine the truth or falsehood of all propositions that can be stated in the terminology of this structure.

A *truth table* is very convenient for showing relationships between several propositions. The fundamental truth tables for conjunction, disjunction, implication, equivalence, and negation are collected together in Table 2.4, in which symbol T means that the corresponding proposition is true, and symbol F means that it is false.

The fundamental axioms of traditional propositional logic are: (1) every proposition is either true or false, but not both true or false; (2) the expressions

[32]Much of the rest of this section is paraphrased from Allendoerfer and Oakley (1955).

Table 2.4 Truth table for five operations that are frequently applied to propositions (Mendel 1995a © 1995, IEEE)

p	q	$p \wedge q$	$p \vee q$	$p \to q$	$p \leftrightarrow q$	$\sim p$
T	T	T	T	T	T	F
T	F	F	T	F	F	F
F	T	F	T	T	F	T
F	F	F	F	T	T	T

given by defined terms are propositions; and (3) the truth Table 2.4 for conjunction, disjunction, implication, equivalence, and negation. Using truth tables, one can derive many interpretations of the preceding operations and can also prove relationships about them.

A *tautology* is a proposition formed by combining other propositions ($p, q, r, \ldots$) which is true regardless of the truth or falsehood of $p, q, r, \ldots$. The most important tautology for our work is:

$$(p \to q) \leftrightarrow \sim [p \wedge (\sim q)] \tag{2.118}$$

A proof of this tautology, using truth tables, is given in Table 2.5. Observe that the entries in the two columns $p \to q$ and $\sim [p \wedge (\sim q)]$ are identical, which proves the tautology. This tautology can also be expressed as

$$(p \to q) \leftrightarrow (\sim p) \vee q \tag{2.119}$$

the truth of which is also demonstrated in Table 2.5. The importance of these tautologies is that they let one express the MF for $p \to q$ in terms of MFs of either propositions p and $\sim q$ or $\sim p$ and q, which is very important for transitioning from crisp to fuzzy logic.

Some of the most important mathematical equivalences between logic and set theory are:

Logic	Set theory
$\wedge$	$\cap$
$\vee$	$\cup$
$\sim$	$\overline{()}$

Table 2.5 Proofs of $(p \to q) \leftrightarrow \sim [p \wedge (\sim q)]$ and $(p \to q) \leftrightarrow (\sim p) \vee q$ (Mendel 1995a © 1995, IEEE)

p	q	$p \to q$	$\sim q$	$p \wedge (\sim q)$	$\sim [p \wedge (\sim q)]$	$\sim p$	$(\sim p) \vee q$
T	T	T	F	F	T	F	T
T	F	F	T	T	F	F	F
F	T	T	F	F	T	T	T
F	F	T	T	F	T	T	T

Table 2.6 Validations of (2.120) and (2.121) (Mendel 1995a © 1995, IEEE)

$\mu_p(x)$	$\mu_q(y)$	$1-\mu_p(x)$	$1-\mu_q(y)$	$1-\min[\mu_p(x), 1-\mu_q(y)]$	$\max[1-\mu_p(x), \mu_q(y)]$
1	1	0	0	1	1
1	0	0	1	0	0
0	1	1	0	1	1
0	0	1	1	1	1

Additionally, as mentioned earlier, there is a correspondence between elementary logic and Boolean Algebra (0, 1). Any statement that is true in one system becomes a true statement in the other, simply by carrying through the following changes in notation:

Logic	Boolean algebra (0, 1)
T	1
F	0
$\wedge$	$\times$
$\vee$	$+$
$\sim$	$'$
$\leftrightarrow$	$=$
$p, q, r, \ldots$	$a, b, c, \ldots$

In this list, $'$ stands for complement, and *a, b, c,* ... are arbitrary elements of the two-element set {0, 1}.

Using the facts that $(p \to q) \leftrightarrow \sim[p \wedge (\sim q)]$ and $(p \to q) \leftrightarrow (\sim p) \vee q$, and the equivalence between logic and set theory, two MFs can be obtained for $p \to q$. The first of these tautologies lets us show that

$$\mu_{p\to q}(x,y) = 1 - \mu_{p\cap\bar{q}}(x,y) = 1 - \min\left[\mu_p(x), 1-\mu_q(y)\right] \tag{2.120}$$

and the second of these tautologies lets us show that[33]

$$\text{Kleene-Dienes}: \mu_{p\to q}^{KD}(x,y) = \mu_{\bar{p}\cup q}(x,y) = \max\left[1-\mu_p(x), \mu_q(y)\right] \tag{2.121}$$

To validate the truth of these two MFs, construct a Boolean truth table, such as the one in Table 2.6. Observe that the entries in the last two columns agree with the entries in Table 2.4 for $p \to q$, where the logical *T* and *F* are interchanged with Boolean 1 and 0, respectively.

[33] A named implication MF (e.g., Kleene-Dienes) refers to the person or persons attributed to it in Klir and Yuan (1995, Table 11.1, p. 309).

The implication MFs given in (2.120) and (2.121) are by no means the only ones that give agreement with $p \to q$. Two others are shown here [see Klir and Yuan (1995, Table 11.1) for many more]:

$$\text{Reichenbach} : \mu^{R}_{p \to q}(x, y) = 1 - \mu_p(x)[1 - \mu_q(y)] \tag{2.122}$$

and

$$\text{Lukasiewicz} : \mu^{L}_{p \to q}(x, y) = \min[1, 1 - \mu_p(x) + \mu_q(y)] \tag{2.123}$$

The MF in (2.122) is similar to the one in (2.120), except that a *product* operation is used for conjunction instead of the minimum operation.

In logic, an *inference rule* is a logical form consisting of a function that takes premises, analyzes their syntax, and returns a conclusion. In traditional propositional (crisp) logic there are two very important inference rules, *Modus Ponens* and *Modus Tollens*:

Modus Ponens:

Premise: x is A
Implication: IF x is A THEN y is B
Consequence: y is B.

Modus Ponens is associated with the implication "A implies B" ($A \to B$). In terms of propositions p and q, Modus Ponens is expressed as $(p \wedge (p \to q)) \to q$.

Modus Tollens:

Premise: y is not B
Implication: IF x is A THEN y is B
Consequence: x is not A.

In terms of propositions p and q, Modus Tollens is expressed as $(\overline{q} \wedge (p \to q)) \to \overline{p}$.

Whereas Modus Ponens plays a central role in engineering applications of logic, due in large part to cause and effect, Modus Tollens does not seem to have yet played much of a role.

2.16 From Crisp Logic to Fuzzy Logic

Fuzzy logic is a type of logic that includes more than just true or false values. It is the logic that deals with situations where one cannot give a clear yes/no (true/false) answer. In fuzzy logic, propositions are represented with *degrees of truthfulness or falsehood*, i.e., fuzzy logic uses a continuous range of truth values in the interval [0, 1] rather than just true or false values. In fuzzy logic, Aristotle's laws of the Excluded Middle and Contradiction are usually broken.

Fuzzy logic begins by borrowing notions from crisp logic, just as fuzzy set theory borrows from crisp set theory. As in our extension of crisp set theory to fuzzy set theory, our extension of crisp logic to fuzzy logic is made by replacing the bivalent MFs of crisp logic with their fuzzy MFs. That is all there is to it; hence, the IF–THEN statement "IF x is A, THEN y is B," where $x \in X$ and $y \in Y$, has a MF $\mu_{A \to B}(x, y)$ where $\mu_{A \to B}(x, y) \in [0, 1]$. Note that $\mu_{A \to B}(x, y)$ measures the degree of truth of the implication relation between x and y, and it resides in the Cartesian product space $X \times Y$. Examples of such MFs are:

$$\mu_{A \to B}(x, y) = 1 - \min[\mu_A(x), 1 - \mu_B(y)] \tag{2.124}$$

$$\mu_{A \to B}^{KD}(x, y) = \max[1 - \mu_A(x), \mu_B(y)] \tag{2.125}$$

and

$$\mu_{A \to B}^{R}(x, y) = 1 - \mu_A(x)(1 - \mu_B(y)) \tag{2.126}$$

which, of course, are fuzzy versions of (2.120)–(2.122), respectively.

In fuzzy logic, Modus Ponens is extended to *Generalized Modus Ponens*:

Premise: x is A^*
Implication: IF x is A THEN y is B
Consequence: y is B^*.

Compare Modus Ponens and Generalized Modus Ponens to see their subtle differences, namely, in the latter, fuzzy set A^* is not necessarily the same as rule antecedent fuzzy set A, and fuzzy set B^* is not necessarily the same as rule consequent B.

Example 2.30 (Mendel 1995a) Consider the rule "IF a man is short, THEN he will not make a very good professional basketball player." Here fuzzy set A is *short man*, and fuzzy set B is *not a very good professional basketball player*. Given Premise 1, as "This man is under five feet tall," A^* is the fuzzy set *man under five feet tall*. Clearly $A^* \neq A$; but, A^* is similar to A. The following consequence is now drawn: "He will make a poor professional basketball player." Here B^* is the fuzzy set *poor professional basketball player*, and $B^* \neq B$, although B^* is indeed similar to B.

In crisp logic a rule will be fired only if the premise is exactly the same as the antecedent of the rule, and the result of such rule firing is the rule's actual consequent. In fuzzy logic, on the other hand, a rule is fired so long as there is a non-zero degree of similarity between the premise and the antecedent of the rule, and the result of such rule firing is a consequent that has a non-zero degree of similarity to the rule's consequent.

Generalized Modus Ponens is a fuzzy composition where the first fuzzy relation is merely the fuzzy set A^*. Consequently, using (2.56), $\mu_{B^*}(y)$ is obtained from the following *sup-star composition* (also called the *compositional rule of inference*):

$$\mu_{B^*}(y) = \sup_{x \in X}[\mu_{A^*}(x) \bigstar \mu_{A \to B}(x, y)] \ \ y \in Y \tag{2.127}$$

To help us understand the meaning of (2.127), some examples will be considered. In all these examples the fuzzy set A^* is assumed to be a fuzzy singleton, i.e.,

$$\mu_{A^*}(x) = \begin{cases} 1 & x = x' \\ 0 & x \neq x' \text{ and } \forall x \in X \end{cases} \tag{2.128}$$

In Chap. 3 this will be called a *singleton fuzzifier* and one will learn why it is so popular. For the singleton fuzzifier, (2.127) becomes:

$$\begin{aligned} \mu_{B^*}(y) &= \sup_{x \in X}[\mu_{A^*}(x) \bigstar \mu_{A \to B}(x, y)] \\ &= \sup[\mu_{A \to B}(x', y), 0] = \mu_{A \to B}(x', y) \ \ y \in Y \end{aligned} \tag{2.129}$$

Eq. (2.129) is true regardless of whether one uses minimum or product for $\bigstar$. Observe that for the singleton fuzzifier the supremum operation is very easy to evaluate, because $\mu_{A^*}(x)$ is non-zero at only one point, x'.

Example 2.31 To begin, the result of using (2.129) for $\mu_{A \to B}(x', y)$ in (2.120) is examined, i.e.,

$$\mu_{B^*}(y) = \mu_{A \to B}(x', y) = 1 - \min[\mu_A(x'), 1 - \mu_B(y)] \ \ y \in Y \tag{2.130}$$

A graphical interpretation of this result is given in Fig. 2.17. Starting with $\mu_B(y)$ in (a), $1 - \mu_B(y)$ is computed as shown in (b), and, for the given level of $\mu_A(x')$ shown in (b), $\min[\mu_A(x'), 1 - \mu_B(y)]$, also shown in (b), is then constructed. Note that the level shown for $\mu_A(x')$ in (b) was chosen arbitrarily (by the author), where $\mu_A(x') \in [0, 1]$. Finally, $1 - \min[\mu_A(x'), 1 - \mu_B(y)]$ is constructed, as shown in (c).

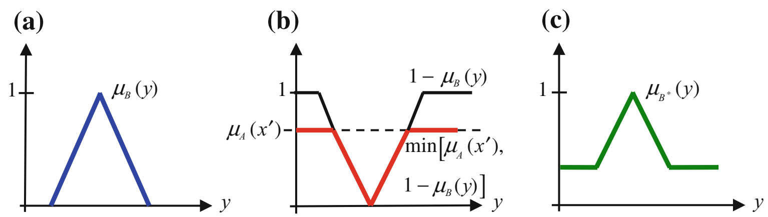


Fig. 2.17 Construction of $\mu_{B^*}(y)$ in (2.130). **a** Consequent MF $\mu_B(y)$, **b** construction of $\min[\mu_A(x'), 1 - \mu_B(y)]$, and **c** $\mu_{B^*}(y)$ (Mendel 1995a © 1995, IEEE)

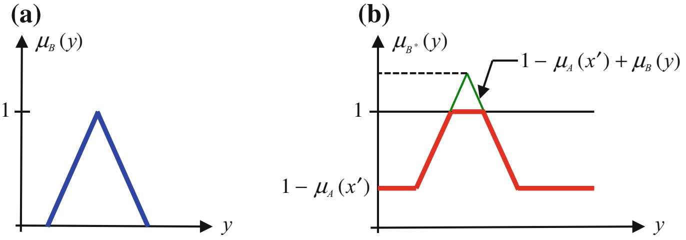


Fig. 2.18 Construction of $\mu_{B^*}(y)$ in (2.132). **a** Consequent MF $\mu_B(y)$, **b** construction of $\mu_{B^*}(y)$ (Mendel 1995a © 1995, IEEE)

The result shown in (c) is disturbing for an engineering application, i.e., given a specific input $x = x'$, the result of firing a specific rule, whose consequent is associated with a specific fuzzy set of finite support [the base of the triangle in (a)], is a fuzzy set whose support is infinite. Somehow a bias (constant) has gotten into the output so that regardless of x' the output is never zero [unless $\mu_A(x') = 1$]. This does not seem desirable for engineering applications.

Example 2.32 Perhaps the problem experienced in Example 2.31 was a result of a poor choice for $\mu_{A\to B}(x', y)$. Therefore, the result of using $\mu^L_{A\to B}(x', y)$ obtained from (2.123), is examined next, i.e.,

$$\mu^L_{A\to B}(x', y) = \min[1, 1 - \mu_A(x') + \mu_B(y)] \;\; y \in Y \tag{2.131}$$

which, by the way, is the implication MF given in the important paper (Zadeh 1973). Substituting this expression for $\mu^L_{A\to B}(x', y)$ into (2.129), it follows that:

$$\mu_{B^*}(y) = \mu^L_{A\to B}(x', y) = \min[1, 1 - \mu_A(x') + \mu_B(y)] \;\; y \in Y \tag{2.132}$$

A graphical interpretation of this result is given in Fig. 2.18. As in Example 2.31, the level shown for $\mu_A(x')$—and subsequently for $1 - \mu_A(x')$—was chosen arbitrarily. Once again, a result has been obtained in Fig. 2.18b that includes a bias. It is easy to demonstrate that all of the other choices provided earlier for $\mu_{A\to B}(x, y)$ have the same problem. Even many choices not listed here have the same problem.

2.17 Mamdani (Engineering) Implications

Mamdani (1974) seems to have been the first one to recognize the problem just demonstrated. Based on simplifying the computations associated with (2.127), he chose to work with the following *minimum implication* (inference)

Table 2.7 Demonstration that minimum and product implications do not agree with $\mu_{p\to q}(x,y)$ (Mendel 1995a © 1995, IEEE)

$\mu_p(x)$	$\mu_q(y)$	$\min[\mu_p(x), \mu_q(y)]$	$\mu_p(x)\mu_q(y)$	$\mu_{p\to q}(x,y)$
1	1	1	1	1
1	0	0	0	0
0	1	0	0	1
0	0	0	0	1

$$\mu_{A\to B}(x,y) \equiv \min[\mu_A(x), \mu_B(y)] \quad x \in X,\ \mathrm{y} \in \mathrm{Y} \tag{2.133}$$

Later, Larsen (1980) proposed the following *product implication* (inference)

$$\mu_{A\to B}(x,y) \equiv \mu_A(x)\mu_B(y) \quad x \in X, y \in Y \tag{2.134}$$

Again, the reason for this choice was simplicity of computation.[34]

Equations (2.133) and (2.134) can be expressed collectively as

$$\mu_{A\to B}(x,y) \equiv \mu_A(x)\bigstar\mu_B(y) \quad x \in X,\ y \in Y \tag{2.135}$$

where ★ is a t-norm, product, or minimum, and is frequently referred to as a *Mamdani implication* regardless of whether the t-norm used is the minimum or product.

Today, minimum and product implications are the most widely used implications in the engineering applications of fuzzy logic; but, what do they have to do with traditional propositional logic? Table 2.7 demonstrates that neither minimum implication nor product implication agrees with the accepted propositional logic definition of implication; hence, minimum and product implications have nothing to do with traditional propositional logic. Consequently, minimum and product implications—Mamdani implications—can be thought of as *engineering implications*.

Because of the use of engineering implication functions in rule-based fuzzy systems and their disconnect from material implication, I now believe it would be better to call such systems "fuzzy systems" rather than "fuzzy logic systems". Hence, *in this book "fuzzy system" is used instead of "fuzzy logic system"*, but "fuzzy system" is not abbreviated to FS, because to do so would confuse it with a fuzzy set.

[34]There is a paragraph in the lower right-hand column on p. 359 of Mendel (1995a) that contains an error. Observe that the derivation of (2.129) has accounted for all values of x, including $x \neq x'$, because it uses (2.128). For some reason that I cannot recall, in the erroneous paragraph, I claim that for all $x \neq x'$, $\mu_{B^*}(y) = 1$, which I then interpret as a form of non-causality, i.e., a rule will be fired for all $x \neq x'$. I then argue for the use of a Mamdani or Larsen implication on the basis of their causality. This is incorrect; however, it does not affect anything else in the 1995 tutorial.

Example 2.33 The purpose of this example is to demonstrate that both the minimum and product implications lead to output fuzzy sets that seem quite reasonable from an engineering perspective, in that they only alter the shape of $\mu_B(y)$ and do not introduce a bias. As in Examples 2.31 and 2.32, singleton fuzzification is assumed, i.e., $\mu_{A^*}(x)$ is given by (2.128).

Considering minimum implication first, (2.129) becomes

$$\mu_{B^*}(y) = \min[\mu_A(x'), \mu_B(y)]\; y \in Y \tag{2.136}$$

A graphical interpretation of this result is given in Fig. 2.19. As in those earlier examples, the level shown for $\mu_A(x')$ was chosen arbitrarily. Observe from Fig. 2.19b that given a specific antecedent $x = x'$ the result of firing a specific rule is a fuzzy set whose support is finite and whose shape is a clipped version of $\mu_B(y)$.

Considering the product implication next, (2.129) becomes:

$$\mu_{B^*}(y) = \mu_A(x')\mu_B(y)\; y \in Y \tag{2.137}$$

A graphical interpretation of this result is given in Fig. 2.20. Similar conclusions are drawn from this figure as were drawn for minimum implication. In this case, the shape of the fuzzy output set is a scaled (attenuated) version of $\mu_B(y)$.

Overall conclusions are that minimum and product implications are, indeed, useful engineering implications, and, that $\mu_{B^*}(y)$ can be expressed as

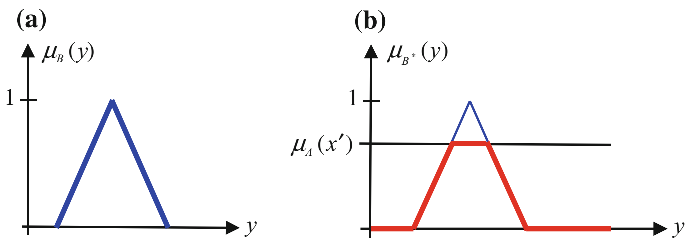


Fig. 2.19 Construction of $\mu_{B^*}(y)$ in (2.136). **a** Consequent MF $\mu_B(y)$, **b** construction of $\mu_{B^*}(y)$ (Mendel 1995a © 1995, IEEE)

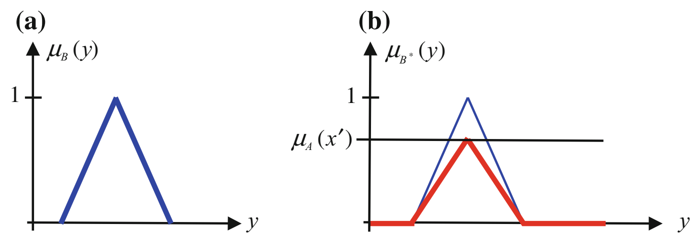


Fig. 2.20 Construction of $\mu_{B^*}(y)$ in (2.137). **a** Consequent MF $\mu_B(y)$, **b** construction of $\mu_{B^*}(y)$ (Mendel 1995a © 1995, IEEE)

$$\mu_{B^*}(y) = \mu_A(x')\bigstar\mu_B(y)\; y \in Y \tag{2.138}$$

where $\bigstar$ is either the minimum or product.

Example 2.34 When there is some uncertainty about the measurement of input variable x, then the measurement can be modeled as a type-1 fuzzy number (in Chap. 3 this will be called a *non-singleton fuzzifier*). Let the measured value of x be denoted x'. In this example a type-1 fuzzy number is created that is centered about x' by using the following Gaussian MF for A^*:

$$\mu_{A^*}(x) = \exp\left(-[(x - x')/\sigma_{A^*}]^2\Big/2\right) \tag{2.139}$$

Here only a single antecedent rule is considered, one whose antecedent MF is also assumed to be a Gaussian, namely:

$$\mu_A(x) = \exp\left(-[(x - m_A)/\sigma_A]^2\Big/2\right) \tag{2.140}$$

Mamdani product implication and product t-norm are assumed, and the goal here is to evaluate the sup-star (product) composition in (2.127).

First, it is shown that the sup-star composition in (2.127) can be expressed as $(y \in Y)$

$$\mu_{B^*}(y) = \left(\sup_{x\in X}[\mu_{A^*}(x)\mu_A(x)]\right) \times \mu_B(y) \tag{2.141}$$

Using product implication, $\mu_{A\to B}(x,y) = \mu_A(x)\mu_B(y)$, and using product t-norm $\bigstar = \times$, (2.127) becomes:

$$\mu_{B*}(y) = \sup_{x\in X}[\mu_{A*}(x)\mu_A(x)\mu_B(y)] = \left(\sup_{x\in X}[\mu_{A*}(x)\mu_A(x)]\right) \times \mu_B(y) \tag{2.142}$$

which is (2.141).

Next, the value of x is established where $\sup_{x\in X}[\mu_{A^*}(x)\mu_A(x)]$ occurs. Let $f(x) \equiv \mu_{A*}(x)\mu_A(x)$, and substitute the Gaussian MFs given in (2.139) and (2.140) into it, to see that

$$f(x) = \exp\left\{-\frac{1}{2}\left[\left(\frac{x - x'}{\sigma_{A*}}\right)^2 + \left(\frac{x - m_A}{\sigma_A}\right)^2\right]\right\} \equiv \exp\left\{-\tfrac{1}{2}\varphi(x)\right\} \tag{2.143}$$

To maximize $f(x)$, $\varphi(x)$, must be minimized; hence, one proceeds as follows:

$$\frac{\partial\varphi(x)}{\partial x} = 2\left(\frac{x - x'}{\sigma_{A*}^2}\right) + 2\left(\frac{x - m_A}{\sigma_A^2}\right) \tag{2.144}$$

Note that $\partial^2\varphi(x)/\partial x^2 = 2/\sigma_{A*}^2 + 2/\sigma_A^2 > 0$; hence, setting $\partial\phi(x)/\partial x = 0$ leads to the value of x that *minimizes* $\varphi(x)$, and subsequently maximizes $f(x)$, i.e.: $\partial\varphi(x)/\partial x = 0 \Rightarrow x = x_{\max}$, which leads to $(x_{\max} - x')\sigma_A^2 + (x_{\max} - m_A)\sigma_{A*}^2 = 0$, from which it is straightforward to show that

$$x_{\max} = \frac{\sigma_{A^*}^2 m_A + \sigma_A^2 x'}{\sigma_A^2 + \sigma_{A^*}^2} \tag{2.145}$$

Finally, $f(x_{\max}) = \sup_{x\in X}[\mu_{A^*}(x)\mu_A(x)]$ is computed. Substitute $x = x_{\max}$ into $\mu_{A*}(x)\mu_A(x)$ and use the middle part of (2.143), to obtain:

$$\begin{aligned} f(x_{\max}) &= \sup_{x\in X}[\mu_{A^*}(x)\mu_A(x)] = \mu_{A^*}(x_{\max})\mu_A(x_{\max}) \\ &= \exp\left\{-\frac{1}{2}\left[\left(\frac{x_{\max} - x'}{\sigma_{A^*}}\right)^2 + \left(\frac{x_{\max} - m_A}{\sigma_A}\right)^2\right]\right\} \end{aligned} \tag{2.146}$$

Using (2.145), it follows that:

$$\frac{x_{\max} - x'}{\sigma_{A*}} = \frac{\sigma_{A*}^2 m_A + \sigma_A^2 x' - (\sigma_A^2 + \sigma_{A*}^2)x'}{(\sigma_A^2 + \sigma_{A*}^2)\sigma_{A*}} = \frac{\sigma_{A*}(m_A - x')}{(\sigma_A^2 + \sigma_{A*}^2)} \tag{2.147}$$

$$\frac{x_{\max} - m_A}{\sigma_A} = \frac{\sigma_{A*}^2 m_A + \sigma_A^2 x' - (\sigma_A^2 + \sigma_{A*}^2)m_A}{(\sigma_A^2 + \sigma_{A*}^2)\sigma_A} = \frac{\sigma_A(x' - m_A)}{(\sigma_A^2 + \sigma_{A*}^2)} \tag{2.148}$$

Consequently,

$$\begin{aligned} f(x_{\max}) &= \exp\left\{-\frac{1}{2}\left[\frac{\sigma_{A*}^2(m_A - x')^2 + \sigma_A^2(x' - m_A)^2}{(\sigma_A^2 + \sigma_{A*}^2)^2}\right]\right\} \\ &= \exp\left\{-\frac{1}{2}\left[\frac{(x' - m_A)^2}{(\sigma_A^2 + \sigma_{A*}^2)}\right]\right\} = f(x') \end{aligned} \tag{2.149}$$

Observe that $f(x_{\max})$ depends on the measured value of x, x', and so it can be treated as a function of x'. Observe, also, that $f(x')$ is also a Gaussian function, one that is centered about m_A and has a variance that is equal to $\sigma_A^2 + \sigma_{A*}^2$; hence, this Gaussian is more spread out than either $\mu_{A^*}(x)$ or $\mu_A(x)$. Once can therefore conclude that the effect of uncertainty on the measured input is to spread out the antecedent's MF.

Exercise 2.40 asks the reader to repeat these computations for Mamdani minimum implication and the minimum t-norm.

2.18 Remarks

So far, all discussions about rules have been for rules with single antecedents, e.g., IF x is A, THEN y is B. Chap. 3 and later chapters describe and characterize rules that have more than one antecedent, e.g.,

$$\text{IF } x_1 \text{ is } F_1 \text{ and } x_2 \text{ is } F_2 \text{ and } \ldots \text{ and } x_p \text{ is } F_p, \text{ THEN } y \text{ is } G$$

In such a multiple-antecedent rule, $x_1 \in X_1, \ldots, x_p \in X_p$, $y \in Y$, and $F_1, \ldots, F_p$ and G are fuzzy sets.

Some other topics, which appear frequently in the fuzzy set literature and are sometimes used in engineering applications of fuzzy set and logic, include: cardinality, similarity and subsethood. Because none of them are used in this book, although they could be used in other applications of rule-based systems, they are left for Exercises 2.43, 2.44 and 2.45, respectively.

The different t-norms, t-conorms, and complements that are available from fuzzy set theory provide some (tough) choices that have to be made in a fuzzy system. Zimmerman (1991, pp. 42–43) describes eight criteria that might be helpful in selecting the connective's operator. Unfortunately, I found most of those criteria to be so subjective that I could not use them in my engineering applications of fuzzy sets.

It is very difficult to make a decision about which t-norm or t-co-norm to use in the fuzzy domain because usually different numerical values are obtained for each choice. It is only back in the crisp domain where the same numerical values are obtained for the different choices that one can make a choice based on complexity (simplicity[35]) of the choice. Interestingly, Zadeh seems to only use the minimum or product for conjunction and the maximum for disjunction, the least complex choices.

Most rule-based engineering applications of fuzzy sets use: (1) the minimum or algebraic product t-norm for fuzzy intersection, (2) the maximum t-conorm for fuzzy union, and (3) $1 - \mu_A(x)$ for the MF of the fuzzy complement. These choices are adhered to in this book.

Finally, I want to comment on fuzzy sets and probability.[36] Some people maintain that there is no difference between fuzzy sets and probability. When I am asked about this, often at the beginning of a lecture or course on fuzzy sets and systems, I ask the following question: "How many of you have had a formal course on probability?" Usually, all hands go up. Then I ask: "How many of you have had a formal course on fuzzy sets and systems." Usually, no hands, or only a very small number of hands go up. I then state that in order to explain the differences between fuzzy sets and probability, one must first spend time formally understanding fuzzy

[35]This is based on Ockham's razor principle; see footnote 13 in Chap. 6 (page 272) for a discussion about this principle.

[36]The rest of the material in this section is taken for the most part from Mendel (1995a).

sets. Only then can intelligent comparisons be made between that which one understands (probability) and that which one will understand (fuzzy sets).

Having just read this chapter, fuzzy sets and probability can now be discussed intelligently.

A lot has been written about fuzzy sets and their relation to probability [e.g., (Cheeseman 1988; Kosko 1990; Laviolette and Seaman 1994; Lindley 1982) and, *IEEE Trans. on Fuzzy Systems*, March 1994, Special Issue]. Many fuzzy set theorists maintain that fuzzy sets are quite different than probability, for a wide variety of reasons, including the facts that: the laws of excluded middle and contradiction are broken for fuzzy sets, but are not broken in probability, and, that conditional probability must be defined in probability theory, but can be derived from first principles using fuzzy sets (Kosko 1990, 1992). Others maintain that fuzzy sets subsume probability. Subjective (as distinguished from frequency-based) probabilists on the other hand, maintain that anything one can do with fuzzy sets can also be done with subjective probability, and that the latter is to be preferred because it has an axiomatic basis, whereas fuzzy sets do not. They bemoan the fact that engineers, who are the largest users of fuzzy systems, are not adequately trained in subjective probability.

The fact of the matter is that there is some truth to both sides of *fuzziness versus probability*. While it is of great intellectual interest to establish the proper connections between fuzzy sets and probability, this author does not believe that doing so will change the ways in which one solves problems, because both probability and fuzzy sets should be in the arsenal of tools used by engineers. Fuzzy sets will not solve all problems, nor will probability.

That fuzzy sets are a tool of enrichment and not replacement is clearly explained in Bezdek and Pal (1992) who ask the question: "Where do fuzzy models fit in with other models?" They then give the following answer (Bezdek and Pal 1992, © IEEE 1992):

> Fuzzy models belong wherever they can provide collateral or competitively better information about a physical process. … we note that each of the following disciplines provides some information about the dynamics of motion: Newtonian mechanics, relativistic mechanics, statistical mechanics, quantum mechanics, and auto mechanics. These models provide us with different, useful, auxiliary, and sometimes contradictory information about various facets of dynamics. Each contributes something about the physical world, so it is with various classes of models. … From a different point of view, because every hard set is fuzzy but not conversely, the mathematical embedding of conventional set theory into fuzzy sets is as natural as the idea of embedding the real numbers into the complex plane. In both cases we can expect the larger 'space' to contain answers to (real) questions that cannot be found in the smaller one. Thus the idea of fuzziness is one of enrichment not of replacement.

Addressing the fuzziness versus probability issue, Bezdek and Pal also ask: "Isn't fuzziness just a clever disguise for probability?" Their answer is ([5], © IEEE 1992):

> … an emphatic no. There is a strong philosophical argument against regarding fuzziness as the surrogate for (frequency-based) probability. The *spirit* of this argument is contained in (the following) example. Let L = set of all liquids, and let fuzzy subset L = {all (*potable*)

liquids}. Suppose you had been in the desert for a week without a drink and you came upon two bottles marked C and A [bottle C is labeled $\mu_L(C) = 0.91$ and bottle A is labeled $\Pr[A \in L] = 0.91$]. Confronted with this pair of bottles, and given that you must drink from the one you choose, which would you choose to drink from? Most readers when presented with this experiment immediately see that while C could contain, say, swamp water, it would not … contain liquids such as hydrochloric acid. That is *membership* of 0.91 means that the contents of C are fairly similar to perfectly potable liquids (e.g., pure water). On the other hand, the probability that A is potable = 0.91 means that over a long run of experiments, the contents of A are expected to be potable in about 91% of the trials; in the other 9% the contents will be deadly—about a 1 chance in 10. Thus, most subjects will opt for a chance to drink swamp water. … There is another facet to this example, and it concerns the idea of *observation*. Continuing then, suppose we examine the contents of C and A and discover them to be Dixie beer and hydrochloric acid, respectively. Note that, *after observation*, the membership value of C is unchanged while the probability value for A drops from 0.91 to 0.0. This example shows that these two models possess philosophically different kinds of information: fuzzy memberships, which represent similarities of objects to imprecisely defined properties; and probabilities, which convey information about relative frequencies.

Appendix 1: Properties of Type-1 Fuzzy Sets

This appendix presents details about properties/laws of type-1 fuzzy sets and examines the following frequently used laws to see if they remain satisfied under maximum t-conorm and either minimum or product t-norms:

Reflexive, anti-symmetric, transitive, idempotent, commutative, associative, absorption, distributive, involution, De Morgan's, and identity

Our reason for doing this is that rules in a rule-based system may make use of the words "and", "or", "unless", "not", etc., but all of the mathematics for such a system is worked out in this book only for canonical rules that use the words "and" and "or". Section 3.2 shows how the former rules can be transformed into the canonical rules by using some of the above laws. So, it is important to know when or if the use of these laws is correct.

The exact nature of all the preceding laws is given in the second column of Table 2.8. These laws are all satisfied for crisp sets (for the minimum and product t-norms), due to the facts that: $\min(0,0) = 0$ and $0 \times 0 = 0$, $\min(1,0) = 0$ and $1 \times 0 = 0$, $\min(0,1) = 0$ and $0 \times 1 = 0$, and, $\min(1,1) = 1$ and $1 \times 1 = 1$. That they are all satisfied for maximum t-conorm and minimum t-norm (a so-called "dual t-conorm and t-norm pair") is well known (e.g. Klir and Yuan 1995) and proofs for this situation are left to the reader (Exercise 2.41).

The rest of this appendix focuses on the maximum t-norm and product t-norm pairing. Reflexive, anti-symmetric, and transitive laws do not make use of any t-norm; hence, they are automatically satisfied for maximum t-conorm and product t-norm. Commutative and associative laws are also satisfied, because both maximum and product operations are commutative and associative; i.e., for $x \in X$ ($\vee \equiv$ maximum):

Table 2.8 Summary of set-theoretic laws and whether or not they are satisfied for type-1 fuzzy sets under maximum t-conorm and either minimum or product t-norms[a]

Set theoretic laws		Minimum t-norm	Product t-norm
Reflexive	$\mu_A \le \mu_A$	Yes	Yes
Anti-symmetric	$\mu_A \le \mu_B, \mu_B \le \mu_A \Rightarrow \mu_A = \mu_B$	Yes	Yes
Transitive	$\mu_A \le \mu_B, \mu_B \le \mu_C \Rightarrow \mu_A \le \mu_C$	Yes	Yes
Idempotent	$\mu_A \vee \mu_A = \mu_A$	Yes	Yes
	$\mu_A \bigstar \mu_A = \mu_A$	Yes	**NO**
Commutative	$\mu_A \vee \mu_B = \mu_B \vee \mu_A$	Yes	Yes
	$\mu_A \bigstar \mu_B = \mu_B \bigstar \mu_A$	Yes	Yes
Associative	$(\mu_A \vee \mu_B) \vee \mu_C = \mu_A \vee (\mu_B \vee \mu_C)$	Yes	Yes
	$(\mu_A \bigstar \mu_B) \bigstar \mu_C = \mu_A \bigstar (\mu_B \bigstar \mu_C)$	Yes	Yes
Absorption	$\mu_A \bigstar (\mu_A \vee \mu_B) = \mu_A$	Yes	**NO**
	$\mu_A \vee (\mu_A \bigstar \mu_B) = \mu_A$	Yes	Yes
Distributive	$\mu_A \bigstar (\mu_B \vee \mu_C) = (\mu_A \bigstar \mu_B) \vee (\mu_A \bigstar \mu_C)$	Yes	Yes
	$\mu_A \vee (\mu_B \bigstar \mu_C) = (\mu_A \vee \mu_B) \bigstar (\mu_A \vee \mu_C)$	Yes	**NO**
Involution	$\mu_{\bar{\bar{A}}} = \mu_A$	Yes	Yes
De Morgan's Laws	$\overline{\mu_A \vee \mu_B} = \mu_{\bar{A}} \bigstar \mu_{\bar{B}}$	Yes	**NO**
	$\overline{\mu_A \bigstar \mu_B} = \mu_{\bar{A}} \vee \mu_{\bar{B}}$	Yes	**NO**
Identity	$\mu_A \vee 0 = \mu_A$	Yes	Yes
	$\mu_A \bigstar 1 = \mu_A$	Yes	Yes
	$\mu_A \vee 1 = 1$	Yes	Yes
	$\mu_A \bigstar 0 = 0$	Yes	Yes

Adapted from Table 1 of Karnik and Mendel (2001)

[a]Arguments of all MFs have been omitted; hence, μ_A, for example, is short for $\mu_A(x)$

$$\mu_A(x) \vee \mu_B(x) = \mu_B(x) \vee \mu_A(x)$$

$$\mu_A(x) \times \mu_B(x) = \mu_B(x) \times \mu_A(x)$$

$$(\mu_A(x) \vee \mu_B(x)) \vee \mu_C(x) = \mu_A(x) \vee (\mu_B(x) \vee \mu_C(x))$$

$$(\mu_A(x) \times \mu_B(x)) \times \mu_C(x) = \mu_A(x) \times (\mu_B(x) \times \mu_C(x))$$

Under product t-norm, the second part of the absorption laws is satisfied, because $\mu_A(x) \times \mu_B(x) \le \mu_A(x)$, so that $\mu_A(x) \vee (\mu_A(x) \times \mu_B(x)) = \mu_A(x)$. The first part of the distributive laws is satisfied; i.e., product is distributive over maximum. The first part of the idempotent laws is also satisfied; i.e., $\mu_A(x) \vee \mu_A(x) = \mu_A(x)$. The involution law is satisfied, since complement is defined as $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$. And, all the identity laws are satisfied (i.e., $\mu_A(x) \vee 0 = \mu_A(x)$, $\mu_A(x) \times 1 = \mu_A(x)$, $\mu_A(x) \vee 1 = 1$, and $\mu_A(x) \times 0 = 0$).

None of the other laws are satisfied under product t-norm, because:

- Idempotent laws—second part

$$\mu_A(x) \times \mu_A(x) \neq \mu_A(x) \tag{2.150}$$

- Absorption laws—first part: assume, e.g. that $\mu_A(x) > \mu_B(x)$; then,

$$\mu_A(x) \times (\mu_A(x) \vee \mu_B(x)) = \mu_A(x) \times \mu_A(x) = \mu_A^2(x) \neq \mu_A(x) \tag{2.151}$$

- Distributive laws–second part: assume, e.g. that $\mu_A(x) > \mu_B(x)$ and $\mu_A(x) > \mu_C(x)$; then,

$$\begin{aligned} \mu_A(x) \vee (\mu_B(x) \times \mu_C(x)) &= \mu_A(x) \\ &\neq (\mu_A(x) \vee \mu_B(x)) \times (\mu_A(x) \vee \mu_C(x)) = \mu_A^2(x) \end{aligned} \tag{2.152}$$

- De Morgan's laws:

$$\begin{aligned} \overline{\mu_A(x) \vee \mu_B(x)} &= 1 - (\mu_A(x) \vee \mu_B(x)) \\ &\neq \mu_{\bar{A}}(x) \times \mu_{\bar{B}}(x) = (1 - \mu_A(x)) \times (1 - \mu_B(x)) \end{aligned} \tag{2.153}$$

$$\begin{aligned} \overline{\mu_A(x) \times \mu_B(x)} &= 1 - \mu_A(x) \times \mu_B(x) \\ &\neq \mu_{\bar{A}}(x) \vee \mu_{\bar{B}}(x) = \max\{(1 - \mu_A(x)), (1 - \mu_B(x))\} \end{aligned} \tag{2.154}$$

Exercises

2.1 Fuzziness as a concept that lets an object reside in more than one set but to different degrees may be traced back to antiquity. Go on the Internet and find a picture of the statue called the *Guardian Sphinx* (530 BC.).

   (a) What are the three sets for this statue?
   (b) What membership grade would you assign to each of the three sets?

2.2 Fuzziness as a concept that lets an object reside in more than one set but to different degrees has occurred in art, even before Zadeh formalized it. For example, it occurs in the works of the Belgian painter Renè Magritte. Go on the Internet and find the following paintings by him and answer the related question:

   (a) *The Explanation* (1952): What is the degree of similarity between the carrot and the wine bottle?
   (b) *Homage to Alphonse Allais* (1964): What is the degree of similarity between the cigar and the fish?

2.3 Suppose that a car is described by its *color*. What scale could be used for color? Create five terms for color and sketch MFs for each term.

2.4 Establish MFs for:

(a) real numbers close to 10
(b) real numbers approximately equal to 6
(c) integers very far from 10
(d) complex numbers near the origin
(e) light (weight)
(f) heavy (weight).

2.5 List six linguistic variables from the field of acoustics (or any field that is of interest to you).

2.6 Using the rules in Example 2.5 as illustrations, list four more rules and their associated MFs.

2.7 Let X be the set of all men and Y be the set of all women. Consider the linguistic variable "weight," and the set of terms {*very skinny*, *skinny*, *just right*, *heavy*, *very heavy*}. Create MFs for these terms for both men and women.

2.8 Consider the judgments listed here, and assume that they can be mapped onto an interval scale ranging from 0 to 10. Define five fuzzy sets for each of them and sketch what you feel are appropriate MFs for them.

(a) touching
(b) eye contact
(c) smiling
(d) acting witty
(e) flirtation.

2.9 Western logic and thinking has been dominated for the most part by the Aristotelian laws of contradiction and the excluded middle. Eastern thinking has not. Eastern religions and concepts such as the Yin and the Yang (female and male/opposite forces) have caused some to speculate that this is why China and Japan were more receptive to fuzzy logic than were people in the West. For example, it's possible for each of you to reside in Yin and Yang simultaneously, but to different degrees. Explain this in terms of fuzzy sets.

2.10 Prove that, for crisp sets A and B, $\min[\mu_A(x), \mu_B(x)]$ provides the correct MF for intersection, given in (2.12).

2.11 For crisp sets A and B, prove the:

(a) commutative law
(b) associative laws
(c) distributive laws
(d) De Morgan's laws.

2.12 Consider three fuzzy sets, A, B, and C, whose MFs are (unnormalized) Gaussians, i.e., $\mu_A(x) = \exp\left[-\frac{1}{2}(x-3)^2\right]$, $\mu_B(x) = \exp\left[-\frac{1}{2}(x-4)^2\right]$ and $\mu_C(x) = \exp\left[-\frac{1}{2}(x-6)^2\right]$. Sketch each of the following:

(a) $$\mu_{A\cap B\cap C}(x)$$

(b) $$\mu_{A\cup B\cup C}(x)$$

(c) $$\mu_{(A\cup B)\cap C}(x) \text{ and } \mu_{A\cup(B\cap C)}(x)$$

(d) $$\mu_{(A\cap B)\cup C}(x) \text{ and } \mu_{A\cap(B\cup C)}(x)$$

(e) $$\mu_{\overline{A\cup B\cup C}}(x).$$

2.13 Consider the fuzzy sets A and B, where $\mu_A(x) = \exp\left[-\frac{1}{2}(x-3)^2\right]$ and $\mu_B(x) = \exp\left[-\frac{1}{2}(x-4)^2\right]$.

(a) Sketch $\mu_{A\cup B}(x)$ for the following t-conorms: maximum, algebraic sum, bounded sum and drastic sum. Which t-conorm gives the largest and smallest values for $\mu_{A\cup B}(x)$?
(b) Sketch $\mu_{A\cap B}(x)$ for the following t-norms: minimum, algebraic product, bounded product and drastic product. Which t-norm gives the largest and smallest values for $\mu_{A\cap B}(x)$?

2.14 Using (2.34) and (2.35), show that $\mu_{c\cup s}(u,v)$ and $\mu_{c\cap s}(u,v)$ are given by (2.38) and (2.39), respectively.

2.15 Verify the max-min and max-product composition of the crisp relations for the (3, 3) element of $R_3(U,W)$ in (2.43).

2.16 Consider the fuzzy relations "u is lighter than v" *or* "u is about the same weight as v." Assume that $u \in U$ and $v \in V$ where U and V are discrete universes of discourse, and U has four elements whereas V has six elements.

(a) Pick U and V to use in the rest of this exercise.
(b) Establish MFs for *lighter* and *about the same*, i.e., $\mu_l(u,v)$ and $\mu_{ats}(u,v)$, where the numbers in $\mu_l(u,v)$ and $\mu_{ats}(u,v)$ agree with a comparison of the numbers in U and V.
(c) Compute $\mu_{l\cup ats}(u,v)$.

2.17 Perform all of the calculations needed to obtain $\mu_{comb}(u,w)$ given in (2.54).

2.18 Repeat Example 2.15 using the product t-norm. Compare these results with the ones given in (2.54) which were obtained using the minimum t-norm. Are they significantly different?

2.19 Consider the fuzzy relation "u is lighter than v" on $U\times V$, and the fuzzy relation "v is heavier than w" on $V\times W$. Assume that U, V, and W are discrete universes of discourse, and U has four elements, V has six elements, and W has three elements.

(a) Pick U, V, and W to use in the rest of this exercise.
(b) Establish MFs for *lighter* and *heavier*, i.e., $\mu_l(u,v)$ and $\mu_h(v,w)$, where the numbers in $\mu_l(u,v)$ and $\mu_h(v,w)$ agree with a comparison of the numbers in U, V, and W.
(c) Compute $\mu_{l \circ h}(u,w)$ using minimum t-norm.
(d) Compute $\mu_{l \circ h}(u,w)$ using product t-norm.
(e) Compare the results from (c) and (d).

2.20 Consider the fuzzy relation "u is lighter than v" on $U \times V$. Assume that U and V are discrete universes of discourse, and U has four elements and V has six elements.

(a) Pick U and V to use in the rest of this exercise.
(b) Establish a MF for *lighter*, i.e., $\mu_l(u,v)$, where the numbers in $\mu_l(u,v)$ agree with a comparison of the numbers in U and V.
(c) Construct a MF for the fuzzy set *skinny*, $\mu_{skinny}(u)$, on U.
(d) Compute the composition of "u is skinny" *and* "u is lighter than v", $\mu_{skinny \circ l}(v)$.

2.21 Using the same universe of discourse as in Example 2.17, develop MFs for:

(a) very likely
(b) not-too-likely.

2.22 Suppose that $U = \{-5,-4,-3,-2,-1,0,1,2,3,4,5\}$ and fuzzy set A is characterized by the MF

$$\mu_A(x) = 0.2/-5 + 0.4/-4 + 0.4/-3 + 0.5/-2 + 0.5/ \\ -1 + 0.6/0 + 0.9/1 + 1/2 + 0.8/3 + 0.5/4 + 0.1/5$$

(a) Determine the MF for the fuzzy set B that is associated with $\mu_{f(A)}(y)$ when $y = f(x) = x^3 + 2x^2$.
(b) Determine the MF for the fuzzy set B that is associated with $\mu_{f(A)}(y)$ when $y = |x|$.

2.23 Suppose that $X_1 = \{1,2,3,4\}$ and $X_2 = \{-1,-2,-3,-4\}$, and fuzzy sets A_1 and A_2 are characterized by the following MFs:

$$\mu_{A_1}(x_1) = 0.5/1 + 0.5/2 + 0/3 + 1/4 \text{ and} \\ \mu_{A_2}(x_2) = 1/-1 + 0/-2 + 0.25/-3 + 0.5/-4$$

Determine the MF for the fuzzy set B that is associated with $\mu_{f(A_1A_2)}(y)$, when $y = f(x_1,x_2) = x_1^2 - 2x_2^2$.

2.24 Given the type-1 Gaussian fuzzy set F_i, with mean m_i and standard deviation σ_i, prove that $a_iF_i + b$ is a Gaussian fuzzy set with mean $a_im_i + b$ and standard deviation $|a_i\sigma_i|$. Note that this result does not depend on the kind of t-norm used, since a_i and b are crisp numbers.

2.25 Given n type-1 Gaussian fuzzy sets $F_1, \ldots, F_n$, with means $m_1, \ldots, m_n$ and standard deviations $\sigma_1, \ldots \sigma_n$, as in (2.77), prove that $\sum_{i=1}^n F_i$ is a Gaussian fuzzy set with mean $\sum_{i=1}^n m_i$ and standard deviation Σ'', where

$$\Sigma'' = \begin{cases} \sqrt{\sum_{i=1}^n \sigma_i^2} & \text{if product t-norm is used} \\ \sum_{i=1}^n \sigma_i & \text{if minimum t-norm is used} \end{cases}$$

[Hints: (1) First prove the results for two sets and then for three sets; (2) show that the supremum of the minimum of two Gaussians is reached at their point of intersection lying between their means.]

2.26 Complete part (b) in the proof of Example 2.21.

2.27 In Example 2.22, obtain the comparable results when a_i are positive or negative real numbers.

2.28 Prove (2.81).

2.29 Repeat Example 2.28 but now for $\mu_{A \cap B}(x)$.

2.30 Let[37] $X_i (i = 1, \ldots, n)$ be fuzzy sets with Gaussian MFs, $\mu_{X_i}(x_i) = \exp\left(-[(x_i - c_i)/\sigma_i]^2 \Big/ 2\right)$, and $w_i \geq 0$ be constant weights with $\sum_{i=1}^n w_i = 1$. Using the Extension Principle with the minimum t-norm, prove that $Y_n = \sum_{i=1}^n w_i X_i$ is a fuzzy set with MF $\mu_{Y_n}(y_n) = \exp\left(-\left[y_n - \sum_{i=1}^n w_i c_i\right]^2 \Big/ \left[\sum_{i=1}^n w_i \sigma_i\right]^2\right)$. [Hint: Prove this by using mathematical induction.]

2.31 Let[38] $A = [a, b, c]$ and $B = [p, q, r]$ be two triangle type-1 fuzzy numbers with MF given in (2.103) and (2.104), respectively. Compute the MF of:

(a) $A - B$

(b) $\exp(A)$

(c) $\ln(A)$

(d) $\sqrt{A}$

(e) $(A)^{1/n}$.

2.32 Let[39] $A = [a, b, c]$ and $B = [p, q, r]$ be two positive triangle type-1 fuzzy numbers with MF given in (2.103) and (2.104), respectively. Compute the MF of:

(a) $A \cdot B$

(b) $A \div B$

(c) $(A)^{-1}$.

---

[37]This exercise is adapted from Wang and Mendel (2016).

[38]This exercise is adapted from Dutta et al. (2011).

[39]This exercise is adapted from Dutta et al. (2011).

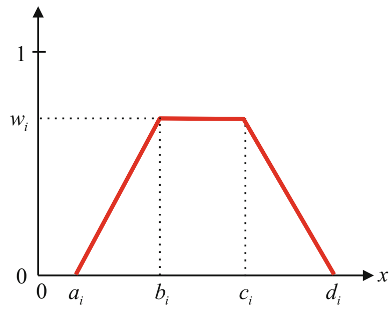


Fig. 2.21 Type-1 trapezoidal fuzzy number for Exercise 2.33

2.33 For the non-normal type-1 trapezoidal fuzzy number, $A_i = (a_i, b_i, c_i, d_i; w_i)$, whose MF is depicted in Fig. 2.21, prove that (Wei and Chen 2009):

(a) $A_1 + A_2 = (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2; \min(w_1, w_2))$, where a_i, b_i, c_i and d_i are real numbers.
(b) $A_1 - A_2 = (a_1 - d_2, b_1 - c_2, c_1 - b_2, d_1 - a_2; \min(w_1, w_2))$, where a_i, b_i, c_i and d_i are real numbers.
(c) $A_1 \cdot A_2 \approx (a_1 \times a_2, b_1 \times b_2, c_1 \times c_2, d_1 \times d_2; \min(w_1, w_2))$, where a_i, b_i, c_i and d_i are positive real numbers.
(d) $A_1/A_2 \approx (a_1/d_2, b_1/c_2, c_1/b_2, d_1/a_2; \min(w_1, w_2))$, where a_i, b_i, c_i and d_i are non-zero positive real numbers.

In (c) and (d), $\approx$ means that the result is a convex type-1 fuzzy set, as in (2.7), in which $g(x)$ and $h(x)$ are not straight lines.

2.34 Using truth tables show that the following are tautologies [3]:

(a) $$p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$$
(b) $$p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$$
(c) $$p \wedge (q \wedge r) \leftrightarrow (p \wedge q) \wedge r$$
(d) $$p \vee (q \vee r) \leftrightarrow (p \vee q) \vee r$$

2.35 Use truth tables to determine whether or not the following propositions are tautologies:

(a) $$(p \wedge q) \rightarrow (p \vee q)$$
(b) $$[(p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee r)] \rightarrow (q \vee s)$$
(c) $$((p \wedge q) \rightarrow r) \leftrightarrow (p \rightarrow r) \vee (q \rightarrow r)$$

2.36 Prove that $[(A \wedge C) \rightarrow D] \wedge [(B \wedge C) \rightarrow D] \leftrightarrow [(A \vee B) \wedge C \rightarrow D]$[Hint: $(p \rightarrow q) \leftrightarrow (\sim p) \vee q$].
2.37 Validate the truth of the crisp implication MFs given in (2.122) and (2.123).
2.38 Repeat Example 2.32 for the following implication MFs, and indicate which of these has a bias in its output:

(a) Kleene-Dienes in (2.121)
(b) Reichenbach in (2.122)

(c) Gödel : $\mu_{A\to B}^{G}(x',y) = \begin{cases} 1 & \mu_{\mathrm{A}}(x') \le \mu_B(y) \\ \mu_{\mathrm{B}}(y) & \mu_A(x') > \mu_B(y) \end{cases}$

(d) Gaines Resher: $\mu_{A\to B}^{GR}(x',y) = \begin{cases} 1 & \mu_A(x') \le \mu_B(y) \\ 0 & \mu_A(x') > \mu_B(y) \end{cases}$

2.39

(a) For the upward sloping lines in Fig. 2.22a, show that the sup-min composition between the lines and the triangle always occurs at the intersection of the line and the right-hand leg of the triangle.
(b) For the downward sloping lines in Fig. 2.22b, show that the sup-min composition between the lines and the triangle always occurs at the intersection of the line and the left-hand leg of the triangle.

2.40 Everything is the same as in Example 2.34, except that in this exercise minimum implication and minimum t-norm are used.

(a) Show that, in this case, the sup-star composition in (2.127) can be expressed as

$$\mu_{B^*}(y) = \min\left[\sup_{x\in X}[\min[\mu_{A^*}(x), \mu_A(x)]], \mu_B(y)\right]$$

(b) Show that $\sup_{x\in X}[\min[\mu_{A^*}(x), \mu_A(x)]]$ occurs at the intersection point of the two Gaussian MFs, namely at

$$x = x_{\max} = (\sigma_{A^*}m_A + \sigma_A x')/(\sigma_{A^*} + \sigma_A).$$

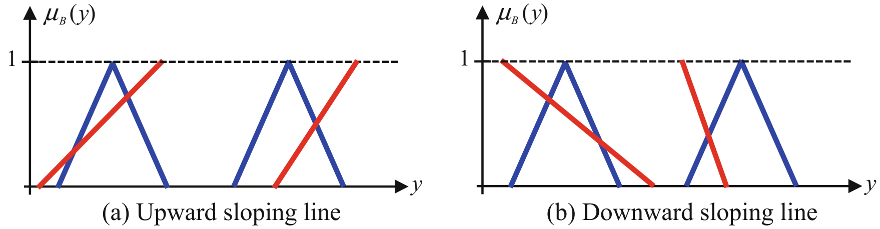


Fig. 2.22 Type-1 fuzzy sets for Exercise 2.39

(c) If possible, obtain a formula for $\sup_{x \in X}[\min[\mu_{A^*}(x), \mu_A(x)]]$.
(d) Assume a Gaussian consequent MF $\mu_B(y)$. Sketch the fired-rule MF $\mu_{B^*}(y)$. How is this obtained directly from sketches of $\mu_{A^*}(x)$, $\mu_A(x)$ and $\mu_B(y)$?
(e) Repeat part (d) for a triangular consequent MF.
(f) Compare the result in part (e) with the result in Fig. 2.19.

2.41 Show that for type-1 fuzzy sets all the set-theoretic laws that are in Table 2.8 are satisfied under maximum t-conorm and minimum t-norm.

2.42 Verify (2.153) and (2.154) numerically.

2.43 As one knows, crisp set A can be defined by using its MF in (2.1). The number of elements that are in A is called its *cardinality*. So, for a crisp set its cardinality can be obtained by summing all of its MF values. Using this idea[40], one can also define the *cardinality of a type-1 fuzzy set* A, $|A|$, analogously (De Luca and Termini 1972), i.e. for a discrete universe, $|A| = \sum_{i=1}^{N} \mu_A(x_i)$, and for a continuous universe, $|A| = \int_X \mu_A(x)dx$. Observe that $|A|$ increases as N increases, and $\lim_{N \to \infty} \sum_{i=1}^{N} \mu_A(x_i)$ does not exist. Wu and Mendel (2007) handle this by defining a *normalized cardinality*, $p(A)$, for a type-1 fuzzy set in which DeLuca and Termini's cardinality definition for continuous universes $|A| = \int_X \mu_A(x)dx$, is discretized, i.e.: $p(A) \equiv \frac{|X|}{N} \sum_{i=1}^{N} \mu_A(x_i)$, where $|X| = x_N - x_1$ is the length of the universe of discourse used in the computation. X can be part of the complete universe of discourse because for some MFs (e.g., Gaussian, Bell) the complete universes of discourse are infinite. Usually x_i $(i = 1, \ldots, N)$ are chosen equally spaced in the domain of x_i; in this case, $p(A)$ converges to its continuous version, $\int_X \mu_A(x)dx$ as N increases.

(a) Compute $|A|$ for the triangle and trapezoidal type-1 fuzzy sets that are in Table 2.3.
(b) Compute $p(A)$ for the same MFs used in (a) for $N = 10, 50, 100$, and compare these results with $|A|$.

2.44 *Similarity* is sometimes used in a rule-based fuzzy system, so this exercise explores similarity for type-1 fuzzy sets. Similarity is about set equality. Two crisp sets A and B are equal if they contain exactly the same elements. In

[40]The wording of the rest of this exercise is taken from Wu and Mendel (2007, p. 5383). The following is also taken from Wu and Mendel (2007, pp. 5382–5383): Definitions of the cardinality of type-1 fuzzy sets have been proposed by several authors, including De Luca and Termini (1972), Kaufman (1977), Gottwald (1980), Zadeh (1981), Blanchard (1982), Klement (1982) and Wygralak (1983). Basically there are two kinds of proposals (Dubois and Prade 1985; Wygralak 2003): (1) those that assume that the cardinality of a type-1 fuzzy set should be a precise number, and (2) those that claim it should be a fuzzy integer. De Luca and Termini's definition of cardinality (also called the *power* of a type-1 fuzzy set) is for the first proposal, is the one that is given in the statement of this exercise, and is the most frequently used definition of cardinality.

crisp set theory either two sets are equal or they are different. For fuzzy sets one knows that everything is a matter of degree; thus for two type-1 fuzzy sets A and B, it is reasonable to define a *degree of similarity*. As usual (in this book), crisp sets are our starting point.

As is stated in Nguyen and Kreinovich (2008): It is known that for two crisp sets A and B: (1) $A \cap B \subseteq A \cup B$ (create a Venn diagram to convince yourself of the truth of this), and (2) $A = B$ iff $A \cap B = A \cup B$. So, for crisp sets, to check whether $A = B$ consider the ratio $|A \cap B|/|A \cup B|$ where $|\bullet|$ denotes the cardinality of $\cdot$ (see Exercise 2.43 about cardinality). In general this ratio is between 0 and 1; the smaller the ratio, the more there are elements from $A \cup B$ which are not part of $A \cap B$, and thus elements from one of the sets A and B that do not belong to the other of these two sets. Thus, for crisp sets, this ratio can be viewed as a reasonable measure of degree to which A is equal to B.

Because there are many definitions of cardinality for a type-1 fuzzy set, and because there can be many ways to define the similarity between two type-1 fuzzy sets (Mendel and Wu 2010 mention that there are at least 50 reported expressions for determining how similar two type-1 fuzzy sets are), this exercise focuses on what is arguably the most popular and useful definition of similarity, the so-called *Jaccard similarity measure*, named after P. Jaccard (Jaccard 1908), who is credited with such a formula.[41] The Jaccard similarity measure, $sm_J(A, B)$, for type-1 fuzzy sets A and B, is: $sm_J(A, B) = f(A \cap B)/f(A \cup B)$. Usually, function f is chosen as the cardinality where $\cap$ = min and $\cup$ = max. For a continuous universe of discourse:

$$sm_J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{\int_X \min(\mu_A(x), \mu_B(x))dx}{\int_X \max(\mu_A(x), \mu_B(x))dx}$$

(a) What is the formula for $sm_J(A, B)$ for discrete universes of discourse?
(b) Compute $sm_J(A, B)$ for the two type-1 fuzzy sets that are depicted in Fig. 2.23a.
(c) Compute $sm_J(A, B)$ for the two type-1 fuzzy sets that are depicted in Fig. 2.23b.

[41]Please note that the use of a crisp number for the similarity of type-1 fuzzy sets is not being absolutely advocated for. Arguments can be given for using a type-1 fuzzy set similarity measure just as well as or for using a crisp number for similarity. The application may dictate which kind of measure is preferable. Of greater importance is that a similarity measure should satisfy some desirable properties, otherwise any kind of a measure between two type-1 fuzzy sets could be claimed to be a similarity measure. Four desirable properties for a type-1 fuzzy set similarity measure $sm(A, B)$ are (e.g., Mendel and Wu 2010, Ch. 4): (1) *Reflexivity*: $sm(A, B) = 1 \Leftrightarrow A = B$; (2) *Symmetry*: $sm(A, B) = sm(B, A)$; (3) *Transitivity*: If $C \le A \le B$ (Note: $A \le B$ if $\mu_A(x) \le \mu_B(x)$ for $x \in X$), where C is an arbitrary fuzzy set on domain X, then $sm(C, A) \ge sm(C, B)$; and (4) *Overlapping*: If $A \cap B \ne \emptyset$, then $sm(A, B) > 0$; otherwise, $sm(A, B) = 0$. $sm_J(A, B)$ satisfies these four properties.

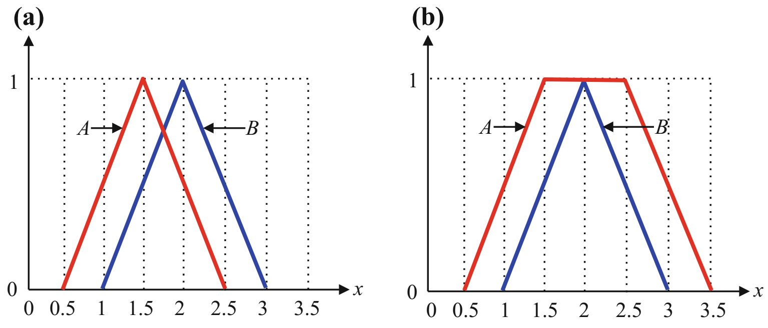


Fig. 2.23 Two type-1 fuzzy sets, A and B, for Exercise 2.44

2.45 *Subsethood* is also sometimes used in a rule-based fuzzy system, so this exercise explores subsethood for type-1 fuzzy sets. Subsethood is about set containment. Containment is dependent on the order of the two sets, A and B, i.e. A can be contained in B but B does not have to be contained in A, e.g. when $A = \{1,2,3\}$ and $B = \{1,2,3,4,5,6\}$, $A \subset B$ but $B \not\subset A$. For crisp sets, it is only when $A = B$ that A is contained in B and B is contained in A. For fuzzy sets one knows that everything is a matter of degree; thus, for two type-1 fuzzy sets A and B, it is reasonable to define a *degree of subsethood*. As usual (in this book), crisp sets are our starting point.

As is stated in Nguyen and Kreinovich (2008): It is known that for two crisp sets A and B: (1) $A \cap B \subseteq A$ and (2) $A \subseteq B$ iff $A \cap B = A$ (create Venn diagrams to convince yourself of the truth of these). So, for crisp sets, to check whether A is a subset of B consider the ratio $|A \cap B|/|A|$ where $|\cdot|$ denotes the cardinality of $\cdot$ (see Exercise 2.43 about cardinality). In general this ratio is between 0 and 1, and it equals 1 if and only if A is a subset of B. The smaller the ratio the more there are elements from A which are not part of the intersection $A \cap B$ and thus not part of set B. Consequently, for crisp sets, this ratio can be viewed as a reasonable measure of the degree to which A is a subset of B (see, also, Kosko 1990, 1992).

Because there are many definitions of cardinality for a type-1 fuzzy set as well as the intersection of two type-1 fuzzy sets, there can be many ways to define the subsethood[42] between two type-1 fuzzy sets. This exercise focuses on what is

[42]Please note that the use of a crisp number for the subsethood of type-1 fuzzy sets is not being absolutely advocated for. Arguments can be given for using a type-1 fuzzy set subsethood measure just as well as or for using a crisp number for subsethood. The application may dictate which kind of measure is preferable. Of greater importance is that a subsethood measure should satisfy some desirable properties, otherwise any kind of a measure between two type-1 fuzzy sets could be claimed to be a subsethood measure. Three desirable properties for type-1 fuzzy set subsethood measure $ss(A,B)$ are (e.g., Mendel and Wu 2010, Ch. 4): (1) *Reflexivity*: $ss(A,B) = 1 \Leftrightarrow A \leq B$ (Note: $A \leq B$ if $\mu_A(x) \leq \mu_B(x)$ for $x \in X$); (2) *Transitivity*: If $C \leq A \leq B$, then $ss(A,C) \geq ss(B,C)$, where C is an arbitrary fuzzy set on domain X, or if $A \leq B$, then $ss(C,A) \leq ss(C,B)$ for any C;

arguably the most widely used definition of subsethood due to Kosko (1990) and denoted here as $ss_K(A,B)$. For a continuous universe of discourse,

$$ss_K(A,B) = \frac{\int_X \min(\mu_A(x), \mu_B(x))dx}{\int_X \mu_A(x)dx}$$

(a) Explain why $ss_K(A,B) \neq ss_K(B,A)$.
(b) What is the formula for $ss_K(A,B)$ for discrete universes of discourse?
(c) Compute $ss_K(A,B)$ for the two type-1 fuzzy sets that are depicted in Fig. 2.23a.
(d) Compute $ss_K(A,B)$ for the two type-1 fuzzy sets that are depicted in Fig. 2.23b.

References

Aisbett, J., J.T. Rickard, and D.G. Morgenthaler. 2010. Type-2 fuzzy sets as functions on spaces. *IEEE Trans on Fuzzy Systems* 18: 841–844.

Allendoerfer, C.B., and C.O. Oakley. 1955. *Principles of mathematics*. New York: McGraw-Hill.

Arabi, B.N., N. Kehtarnavaz, and C. Lucas. 2001. Restrictions imposed by the fuzzy extension of relations and functions. *Journal of Intelligent and Fuzzy Systems* 11: 9–22.

Bezdek, J.C. 1981. *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum.

Bezdek, J., and S.K. Pal. 1992. *Fuzzy models for pattern recognition*. New York: IEEE Press.

Blanchard, N. 1982. Cardinal and ordinal theories about fuzzy sets. In *Fuzzy Information and Decision Process*, ed. M. M. Gupta and E. Sanchez, pp. 149–157, Amsterdam.

Bonissone, P. P. and K. S. Decker. 1986. Selecting uncertainty calculi and granularity: An experiment in trading off precision and complexity. In *Uncertainty in Artificial Intelligence*, ed. L. N. Kanal and J. F. Lemmer, pp. 217–247, Amsterdam.

Cheeseman, P. 1988. An inquiry into computer understanding. *Computational Intelligence* 4: 57–142 (with 22 commentaries/replies).

Cox, E.A. 1994. *The fuzzy systems handbook*. Cambridge, MA: AP Professional.

Cox, E. A. 1992. Fuzzy fundamentals. *IEEE Spectrum*, pp. 58–61, Oct. 1992.

Dubois, D., and H. Prade. 1980. *Fuzzy sets and systems: Theory and applications*. NY: Academic Press.

Dubois, D., and H. Prade. 1985. Fuzzy cardinality and the modeling of imprecise quantification. *Fuzzy Sets and Systems* 16: 199–230.

Dutta, P. H. Boruah and T. Ali. 2011. Fuzzy arithmetic with and without using the α-cut method: A comparative study. *International Journal of Latest Trends in Computing* 2(1): 99–107 (E-ISSN: 2045-5364).

(Footnote 42 continued)

and (3) *Overlapping*: If $A \cap B \neq \emptyset$, then $ss(A,B) > 0$; otherwise, $ss(A,B) = 0$. $ss_K(A,B)$ satisfies these three properties. The interested reader is referred to, e.g. Young (1996) and Fan et al. (1999).

Edwards, W.F. 1972. *Likelihood*. London: Cambridge Univ. Press.

Fan, J.L., W.X. Xie, and J. Pei. 1999. Subsethood measure: new definitions. *Fuzzy Sets and Systems* 106: 201–209.

Gottwald, S. 1980. A note on fuzzy cardinals. *Kybernetika* 16: 156–158.

He, Q., H.-X. Li, C.L.P. Chen, and E.S. Lee. 2000. Extension principles and fuzzy set categories. *Computers and Mathematics with Applications* 39: 45–53.

Horikawa, S., T. Furahashi and Y. Uchikawa. 1992. On fuzzy modeling using fuzzy neural networks with back-propagation algorithm. *IEEE Transaction on Neural Networks* 3: 801–806.

Jaccard, P. 1908. Nouvelles recherches sur la distribution florale. *Bulletin de la Societe de Vaud des Sciences Naturelles* 44: 223, 1908.

Jang, J.-S.R. 1992. Self-learning fuzzy controllers based on temporal back-propagation. *IEEE Transaction on Neural Networks* 3: 714–723.

Jang, L.-C., and D. Ralescu. 2001. Cardinality concepts for type-two fuzzy sets. *Fuzzy Sets and Systems* 118: 479–487.

Jang, J.-S. R., C-T. Sun and E. Mizutani. 1997. *Neuro-Fuzzy and Soft-Computing*, Prentice-Hall, Upper Saddle River, NJ.

Karnik, N.N., and J.M. Mendel. 2001. Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems* 122: 327–348.

Karnik, N. N. and J. M. Mendel. 1998. *An Introduction to Type-2 Fuzzy Logic Systems*, USC-SIPI Report #418, Univ. of Southern Calif., Los Angeles, CA, June 1998. This can be accessed at: http://sipi.usc.edu/research; then choose "sipi technical reports/418.

Kaufmann, A. 1977. *Introduction a la theorie des sous-ensembles flous, complement et nouvelles applications*, vol. 4. Paris: Masson.

Klement, E. P. 1982. On the cardinality of fuzzy sets. *Proceedings of 6th European Meeting on Cybernetics and Systems Research*, Vienna, pp. 701–704.

Klir, G.J., and T.A. Folger. 1988. *Fuzzy sets, uncertainty, and information*. Englewood Cliffs, NJ: Prentice Hall.

Klir, G.J., and B. Yuan. 1995. *Fuzzy sets and fuzzy logic: Theory and applications*. Upper Saddle River, NJ: Prentice Hall.

Kosko, B. 1990. Fuzziness versus probability. *International Journal of General Systems* 17: 211–240.

Kosko, B. 1992. *Neural network and fuzzy systems, A dynamical systems approach to machine intelligence*. Englewood Cliffs, NJ: Prentice-Hall.

Kosko. 1986. Fuzzy entropy and conditioning. *Information Sciences* 40: 165–174.

Kreinovich, V. 2008. Relations between interval computing and soft computing. In *Processing with Interval and Soft Computing*, ed. A. de Korvin and V. Kreinovich, 75–97. Springer, London.

Larsen, P. M. 1980. Industrial applications of fuzzy logic control. *International Journal Man-Machine Studies* 12: 3–10.

Laviolette, M., and J.W. Seaman Jr. 1994. The efficacy of fuzzy representations and uncertainty. *IEEE Transaction on Fuzzy Systems* 2: 4–15.

Lin, C.-T., and C.S.G. Lee. 1996. *Neural fuzzy systems: A neuro-fuzzy synergism to intelligent systems*. Upper Saddle River, NJ: Prentice-Hall PTR.

Lindley, B. Y. 1982. Scoring rules and the inevitability of probability. *International Statistical Review* 50: 1–26 (with 7 commentaries/replies).

De Luca, A., and S. Termini. 1972. A definition of non-probabilistic entropy in the setting of fuzzy sets theory. *Information and Computation* 20: 301–312.

Macvicar-Whelen, P. J. 1978. Fuzzy sets, the concept of height, and the hedge 'very'. *IEEE Transaction on Systems, Man, and Cybernetics* SMC-8: 507–511.

Mamdani, E.H. 1974. Applications of fuzzy algorithms for simple dynamic plant. *Proceedings of the IEEE* 121: 1585–1588.

Mendel, J.M. 1995a. Fuzzy logic systems for engineering: A tutorial. *IEEE Proceedings* 83: 345–377.

Mendel, J.M. 1995b. *Lessons in estimation theory for signal processing*. Prentice-Hall PTR, Englewood Cliffs, NJ: Communications and Control.

Mendel, J.M. 2007. Computing with words: Zadeh, Turing, Popper and Occam. *IEEE Computational Intelligence Magazine* 2: 10–17.

Mendel, J.M. 2015. Type-2 fuzzy sets and systems: A retrospective. *Informatik Spektrum* 38 (6): 523–532.

Mendel, J.M., H. Hagras, W.-W. Tan, W.W. Melek, and H. Ying. 2014. *Introduction to type-2 fuzzy logic control*. Hoboken, NJ: John Wiley and IEEE Press.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: Aiding people in making subjective judgments"*. Hoboken, NJ: Wiley and IEEE Press.

Nguyen, H.T. 1978. A note on the extension principle for fuzzy sets. *Journal of Mathematical Analysis and Applications* 64: 369–380.

Nguyen, H. T. and V. Kreinovich. 2008. Computing degrees of subsethood and similarity for interval-valued fuzzy sets: Fast algorithms. In *Proceedings of* 9th *International Conference on Intelligent Technologies in Tech'08*, pp. 47–55, Samui, Thailand, Oct. 2008.

Rajati, M.R., and J.M. Mendel. 2013. Novel weighted averages versus normalized sums in computing with words. *Information Sciences* 235: 130–149.

Rudin, W. 1966. *Real and complex analysis*. New York: Mc-Graw Hill.

Ruspini, E. 1969. A new approach to clustering. *Information Control* 15: 22–32.

Schmucker, K.S. 1984. *Fuzzy set, natural language computations, and risk analysis*. Rockville, MD: Computer Science Press.

Wang, L.-X. 1997. *A course in fuzzy systems and control*. Upper Saddle River, NJ: Prentice-Hall.

Wang, L.-X., and J.M. Mendel. 2016. Fuzzy opinion networks: A mathematical framework for the evolution of opinions and their uncertainties across social networks. *IEEE Transaction on Fuzzy Systems* 24: 880–905.

Wang, L.-X. and J. M. Mendel. 1992a. Fuzzy basis functions, universal approximation, and orthogonal least squares learning. *IEEE Trans. on Neural Networks* 3: 807–813.

Wang, L.-X. and J. M. Mendel. 1992b. Back-propagation of fuzzy systems as non-linear dynamic system identifiers. *Proceedingas of IEEE Intternational Conference on Fuzzy Systems,* 1409–1418, San Diego, CA.

Wei, S-H. and S-M. Chen. 2009. Fuzzy risk analysis based on interval-valued fuzzy numbers. *Expert Systems with Applications* 36: 2285–2299.

Wu, D., and J.M. Mendel. 2007. Uncertainty measures for interval type-2 fuzzy sets. *Information Sciences* 177: 5378–5393.

Wygralak, M. 1983. A new approach to the fuzzy cardinality of finite fuzzy sets. *Busefal* 15: 72–75.

Wygralak, M. 2003. *Cardinalities of fuzzy sets*. Heidelberg: Springer.

Yager, R.R. 1986. A characterization of the fuzzy extension principle. *Journal Fuzzy Sets and Systems* 18: 205–217.

Yager, R.R., and D.P. Filev. 1994. *Essentials of fuzzy modeling and control*. New York: Wiley.

Young, V.R. 1996. Fuzzy subsethood. *Fuzzy Sets and Systems* 77: 371–384.

Zadeh, L.A. 1965. Fuzzy sets. *Information and Control* 8: 338–353.

Zadeh, L.A. 1971. Similarity relations and fuzzy orderings. *Information Sciences* 3 (2): 177–200.

Zadeh, L.A. 1972. A fuzzy-set-theoretic interpretation of iinguistic hedges. *Journal of Cybernetics* 2: 4–34.

Zadeh, L.A. 1981. Possibility theory and soft data analysis. In *Mathematical frontiers of the social and policy sciences*, ed. L. Cobb, and R.M. Thrall, 69–129. CO: Westview Press, Boulder.

Zadeh, L. A. 1973. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transaction on Systems, Man, and Cybernetics* SMC-3: 28–44.

Zadeh, L. A. 1975. The concept of a linguistic variable and Its application to approximate reasoning–1. *Information Sciences* 8: 199–249.

Zadeh, L. A. 1999. From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions. *IEEE Transaction on Circuits and Systems–I: Fundamental Theory and Applications*, 4: 105–119.

Zimmermann, H.J. 1991. *Fuzzy set theory and its applications*, 2nd ed. Boston, MA: Kluwer Academic Publ.

Chapter 3
Type-1 Fuzzy Systems

3.1 Type-1 Fuzzy Systems

The type-1 (T1) fuzzy systems that are studied in this chapter can all be depicted as in Fig. 3.1, which is the same as Fig. 1.2, except that the Fig. 1.2 "Output Processor" block has been replaced by the "Defuzzifier" block. Because all of the fuzzy systems that are described in this chapter are T1 fuzzy systems, in the rest of this chapter "T1 fuzzy system" is shortened to "fuzzy system. Observe, from Fig. 3.1, that the major components of a fuzzy system are rules, fuzzifier, inference engine, and defuzzifier, each of which is explained in this chapter. Observe, also, that it is a specific value of $\mathbf{x}$, namely $\mathbf{x}'$, that excites the fuzzy system, so that the crisp output y depends on it, as $f(\mathbf{x}')$. Exactly what this nonlinear function is will be established in this chapter.

3.2 Rules

Rules in a fuzzy system can have two different canonical structures, one structure attributed to Zadeh (1973) and the other structure attributed to Takagi and Sugeno (1985) and Sugeno and Kang (1988). Although the structures of the antecedents of these two kinds of rules are identical, it is the structures of their consequents that are very different.[1]

Suppose that a fuzzy system has p inputs $x_1 \in X_1, \ldots, x_p \in X_p$, and one output $y \in Y$, where x_1 is described by Q_1 linguistic terms ($T_{x_1} = \{X_{1j}\}_{j=1}^{Q_1}$), x_2 is described

[1]Most presentations about fuzzy systems have separate treatments for Zadeh and TSK rules, something that was done in the first edition of this book (Mendel 2001); however, in this edition of the book these treatments are unified.

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_3

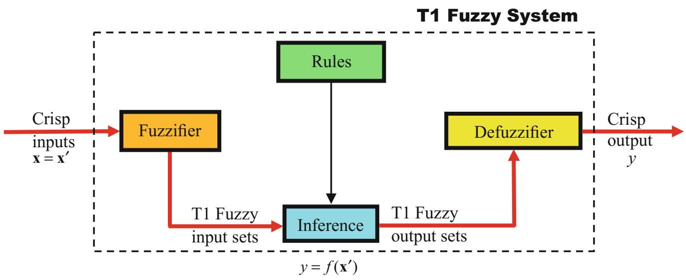


Fig. 3.1 Type-1 fuzzy system. All of its fuzzy sets are type-1 fuzzy sets (T1 FSs)

by Q_2 linguistic terms $(T_{x_2} = \{X_{2j}\}_{j=1}^{Q_2}), \ldots, x_p$ is described by Q_p linguistic terms $(T_{x_p} = \{X_{pj}\}_{j=1}^{Q_p})$, and y is either described by Q_y linguistic terms $(T_y = \{Y_j\}_{j=1}^{Q_y})$, or by a function $g(x_1, \ldots, x_p)$.

Definition 3.1 The structure of the lth generic *Zadeh rule*[2] for a fuzzy system is $(l = 1, \ldots, M)$:

$$R_Z^l: \text{ IF } x_1 \text{ is } F_1^l \text{ and } \cdots \text{ and } x_p \text{ is } F_p^l, \text{ THEN } y \text{ is } G^l \tag{3.1}$$

whereas the structure of the lth generic *Takagi, Sugeno and Kang* (*TSK*, for short) *rule* for a fuzzy system is $(l = 1, \ldots, M)$:

$$R_{TSK}^l: \text{ IF } x_1 \text{ is } F_1^l \text{ and } \cdots \text{ and } x_p \text{ is } F_p^l, \text{ THEN } y \text{ is } g^l(x_1, \ldots, x_p) \quad l = 1, \ldots, M \tag{3.2}$$

In both (3.1) and (3.2), $F_1^l \in T_{x_1}$, $F_2^l \in T_{x_2}$, …, and $F_p^l \in T_{x_p}$. In (3.1), because $G^l \in T_y$ is a type-1 fuzzy set, it is described by its membership function (MF) $\mu_{G^l}(y)$. In (3.2), although y does not seem to be a fuzzy set, it can be modeled as a *type-1 fuzzy singleton* (see Definition 2.2) G^l, so (3.2) is made to resemble a Zadeh rule, where

$$\mu_{G^l}(y) \equiv \begin{cases} 1 & \text{when } y = g^l(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

In (3.3), $\mathbf{x} = col\,(x_1, \ldots, x_p)$.

Both of these generic rules are said to be *complete IF rules* because all p inputs are present in their antecedents. These generic rules represent type-1 fuzzy relations

[2]There are different ways to express a Zadeh rule. The way that it is done in this book is in agreement with Wang (1997).

between the input space $X_1 \times \cdots \times X_p$ and the output space, Y, of the fuzzy system, and which rule structure to use is very much application dependent.

Definition 3.2 When a fuzzy system uses Zadeh rules and a Mamdani implication operator it will be referred to as a *Mamdani fuzzy system*.[3]

Definition 3.3 When a fuzzy system uses TSK rules and a Mamdani implication operator it will be referred to as a *TSK* fuzzy system.

Example 3.1 TSK rules are widely used in fuzzy logic control (FLC); more will be said about FLC in Sect. 4.7. Rules for a discrete-time type-1 TSK fuzzy logic controller often have the following structure $(l = 1, \ldots, M)$:

$$R^l_{TSK} : \text{IF } x(k) \text{ is } F^l_1 \text{ and } x(k-1) \text{ is } F^l_2 \text{ and } \cdots \text{ and } x(k-p+1) \text{ is } F^l_p, \text{ THEN } u_l = c^l_1 x(k) + c^l_2 x(k-1) + \cdots + c^l_p x(k-p+1) \tag{3.4}$$

In (3.4), $x(k)$ and its time-delayed versions are the states of the dynamical system under control, and the consequent is the control u_l that here is a linear combination of the p states.

Rules for a continuous-time type-1 TSK fuzzy logic controller often have the following structure $(l = 1, \ldots, M)$:

$$R^l_{TSK} : \text{IF } x_1(t) \text{ is } F^l_1 \text{ and } x_2(t) \text{ is } F^l_2 \text{ and } \cdots \text{ and } x_p(t) \text{ is } F^l_p, \text{ THEN } u_l = c^l_1 x_1(t) + c^l_2 x_2(t) + \cdots + c^l_p x_p(t) \tag{3.5}$$

In (3.5), $x_i(t)$ denotes the ith state of the dynamical system under control, and the consequent is the control u_l that is also a linear combination of the p states.

It is well known that a multiple-antecedent multiple-consequent rule, in which the multiple consequents are all connected by "or," can always be considered as a group of multi-input single-output rules (Lee 1990, p. 426), which is why only multi-input single-output rules are focused on in this book.

It is also possible to cast "nonobvious" rules [i.e., rules that do not have the exact antecedent structure of the ones in (3.1) and (3.2)] into the forms of the complete rules in (3.1) and (3.2). Six such rules are summarized next, because it is very important for the reader to understand the power and flexibility of the generic rule structures of (3.1) and (3.2). The first five are adapted from Wang (1994, 1997) and are explained next in the context of a Zadeh rule, but the explanations apply equally well for a TSK-rule (except for Comparative Rules), because the antecedent structures of both kinds of rules are the same.

---

[3]This could also be called a *Zadeh fuzzy system*, but people do not seem to use this designation.

1. *Incomplete IF Rules*: Suppose that one has created a rule base where there are p inputs, but some rules have antecedents that are only a subset of the p inputs, e.g.,

$$\text{IF } x_1 \text{ is } \mathrm{F}_1 \text{ and } \cdots \text{ and } x_m \text{ is } \mathrm{F}_m, \text{ THEN } y \text{ is } G.$$

   Such rules are called *incomplete IF rules* and apply regardless of $x_{m+1}, \ldots, x_p$. They can be put into the format of the complete IF rule (3.1) by treating the unnamed antecedents $(x_{m+1}, \ldots, x_p)$ as elements of the fuzzy set INCOMPLETE (IN for short) where, by definition $\mu_{IN}(x) = 1$ for $x \in X$, i.e.,

$$\begin{aligned}&(\text{IF } x_1 \text{ is } F_1 \text{ and } \cdots \text{ and } x_m \text{ is } F_m, \text{ THEN } y \text{ is } G)\\ &\Leftrightarrow (\text{IF } x_1 \text{ is } F_1 \text{ and } \cdots \text{ and } x_m \text{ is } F_m \text{ and } x_{m+1} \text{ is } IN \ \cdots \text{ and } x_p \text{ is } IN\ , \text{ THEN } y \text{ is } G)\end{aligned}$$

2. *Mixed Rules*: Not all rules use the "and" connective; some use the "or" connective and some use a mixture of both. The latter rules are called *mixed rules*. These rules can be decomposed into a collection of equivalent rules, using standard techniques from crisp logic. Suppose, for example, one has the rule:

$$\text{IF } (x_1 \text{ is } F_1 \text{ and } \cdots \text{ and } x_m \text{ is } F_m) \text{ or } (x_{m+1} \text{ is } \mathrm{F}_{m+1} \text{ and } \cdots \text{ and } x_p \text{ is } F_p), \text{ THEN } y \text{ is } G$$

   This rule can be expressed as the following two rules that are connected by "or":

$$\begin{aligned}R_Z^1 &: \text{ IF } x_1 \text{ is } F_1 \text{ and } \cdots \text{ and } x_m \text{ is } F_m, \text{ THEN } y \text{ is } G\\ R_Z^2 &: \text{ IF } x_{m+1} \text{ is } F_{m+1} \text{ and } \cdots \text{ and } x_p \text{ is } F_p, \text{ THEN } y \text{ is } G\end{aligned}$$

   Observe that both of these rules are Incomplete IF rules. See Vadiee and Jamshidi (1993) for related discussions on nesting of rules.

3. *Fuzzy Statement Rules*: Some rules do not appear to have any antecedents; they are statements involving fuzzy sets. Hence, they are called *fuzzy statement rules*. For example, "y is G" is such a rule. Clearly, this is an extreme case of an incomplete IF rule, and can therefore be formulated as:

$$\text{IF } x_1 \text{ is } IN \text{ and } \cdots \text{ and } x_p \text{ is } IN, \text{ THEN } y \text{ is } G$$

4. *Comparative Rules*: Some rules are comparative, e.g.,

   The smaller the x the bigger the y.

   Such rules must first be reformulated as IF–THEN rules; this takes some experience. The preceding rule can be expressed as:

$$\text{IF } x \text{ is } S, \text{ THEN } y \text{ is } B$$

where S is a fuzzy set representing *smaller* and B is a fuzzy set representing *bigger*. Comparative rules may not be appropriate for TSK rules because the latter have functions as consequents rather than words.

5. *Unless Rules*: Rules are sometimes stated using the connective "unless"; such rules are called *unless rules* and can be put into the format of (3.1) by using set theory operations, including De Morgan's Laws. For example, the rule

$$y \text{ is } G \text{ unless } x_1 \text{ is } \mathrm{F}_1 \text{ and } \cdots \text{ and } x_p \text{ is } F_p$$

can first be expressed as

$$\text{IF not}\,(x_1 \text{ is } F_1 \text{ and } \cdots \text{ and } x_p \text{ is } F_p),\ \text{THEN } y \text{ is } G$$

Using De Morgan's Law, $\overline{A \cap B} = \bar{A} \cup \bar{B}$, this rule can be re-expressed as

$$\text{IF } x_1 \text{ is not } F_1 \text{ or } \cdots \text{ or } x_p \text{ is not } F_p,\ \text{THEN } y \text{ is } G$$

"Not F_i" is treated as a fuzzy set and this rule is then decomposed into a collection of p *incomplete IFrules* (connected by "or") each of the form

$$\text{IF } x_i \text{ is not } F_i,\ \text{THEN } y \text{ is } G, \quad i = 1, \ldots, p$$

Although De Morgan's Laws are always valid in crisp set theory and logic, they may not be valid in fuzzy set theory and logic. For example, they are valid for maximum t-conorm and minimum t-norm, but are invalid for maximum t-conorm and product t-norm. See Appendix 1 of Chap. 2 for discussions about this.

6. *Quantifier Rules*: Rules sometimes include the quantifiers "some" or "all"; such rules are called *quantifier rules*. Because of the duality between propositional logic and set theory, rules with the quantifier "some" mean that one has to apply the union operator to the antecedents or consequents to which the "some" applies, whereas rules with the quantifier "all" mean one has to apply the intersection operator to the antecedents or consequents to which the "all" applies.

Of course, in practical applications, it is possible to have rules that combine nonobvious IF-THEN rules 1–6 in all sorts of interesting ways.

If x_1 has Q_1 linguistic terms associated with it, x_2 has Q_2 linguistic terms linguistic terms associated with it, …, and x_p has Q_p linguistic terms associated with it, then a fuzzy system will have a maximum of $Q_1 \cdot Q_2 \cdot \ \cdots \ \cdot Q_p$ rules (i.e., $\max M = Q_1 \cdot Q_2 \cdot \ \cdots \ \cdot Q_p$). The reason that this is $\max M$ and not M is because it may not be necessary to establish all $Q_1 \cdot Q_2 \cdot \ \cdots \ \cdot Q_p$ rules, e.g., it may be known from some expert knowledge that there are regions in $X_1 \times X_2 \times \cdots \times X_p$ where measured values of $x_1, x_2, \ldots, x_p$ do not occur.

When the number of rules is large, one speaks of *rule explosion*. By examining $Q_1 \cdot Q_2 \cdot \cdots \cdot Q_p$, it is clear that rule explosion can occur if rules have many antecedents or if many linguistic terms are used to describe each input. Section 3.9.4 discusses rule explosion in detail.

3.3 Fuzzifier

The *fuzzifier* maps a crisp point $\mathbf{x} = (x_1, \ldots, x_p)^T \in X_1 \times X_2 \times \cdots \times X_p \equiv \mathbf{X}$ into a fuzzy set $A_{\mathbf{x}}$ in $\mathbf{X}$. There are two kinds of fuzzifiers: singleton and non-singleton.

Definition 3.4 A *singleton fuzzifier* is one for which[4] $(i = 1, \ldots, p)$ $\mu_{X_i}(x'_i) = 1$ and $\mu_{X_i}(x_i) = 0$ for $x_i \in X_i$ and $x_i \neq x'_i$.

Definition 3.5 A *non-singleton fuzzifier* maps measurement $(i = 1, \ldots, p)$ $x_i = \mathrm{x}'_i$ into a type-1 fuzzy number (see Definition 2.5) for which $\mu_{X_i}(x'_i) = 1$ and $\mu_{X_i}(x_i)$ decreases from unity as x_i moves away from x'_i . Because the MF for x'_i will be that of a type-1 fuzzy number it is expressed as $\mu_{X_i}(x_i|x'_i)$.

Conceptually, the non-singleton fuzzifier implies that the given input value x'_i is the most likely value to be the correct one from all the values in its immediate neighborhood; however, because the input is uncertain [e.g., it is corrupted by measurement uncertainty (noise)], neighboring points are also likely to be the correct value, but to a lesser degree.

Example 3.2 It is up to the designer to determine the shape of the MF $\mu_{X_i}(x_i|x'_i)$ based on an estimate of the kind and quantity of measurement uncertainty present. It seems logical, though, for this MF to be symmetric about x'_i since the effect of measurement uncertainty is most likely to be equivalent on all points. Examples of such MFs are given below:

1. Gaussian: $\mu_{X_i}(x_i|x'_i) = \exp[-x_i - x'_i)^2/2\sigma^2]$
2. Triangular: $\mu_{X_i}(x_i|x'_i) = \max(0, 1 - |(x_i - x'_i)/c|)$

In these MFs, x'_i is the center value of the fuzzy sets and σ or c is the spread of these sets. Larger values of the spread for these MFs imply that more measurement uncertainty is anticipated to exist in the data.

When $\sigma \rightarrow 0$, or $c \rightarrow 0$, then these non-singleton fuzzifiers reduce to singleton fuzzifiers, because $\mu_{X_i}(x_i|x'_i) = 1$ when $x_i = x'$ and $\mu_{X_i}(x_i|x'_i) = 0$ otherwise. So, singleton fuzzification may be considered a special case of non-singleton

[4]Separable MFs are assumed, as is discussed in Sect. 2.14.

fuzzification. However, because singleton fuzzification is much more widely used than is non-singleton fuzzification, both kinds of fuzzifiers are treated separately in the rest of this chapter.

3.4 Fuzzy Inference Engine

In this section, general results are obtained that are not dependent upon the nature of the fuzzifier, after which those results are specialized to both singleton and non-singleton fuzzifiers.

3.4.1 General Results

In the fuzzy inference engine (which is labeled *Inference* in Fig. 3.1), fuzzy logic principles are used to map fuzzy input sets in $X_1 \times \cdots \times X_p$, that flow through an IF-THEN rule (or set of rules), into fuzzy output sets in Y. Each rule is interpreted as a fuzzy implication.

To begin the focus here is on the Zadeh rules in (3.1) and a Mamdani fuzzy system. Let $F_1^l \times \cdots \times F_p^l = A^l$; then, (3.1) can be re-expressed as $(l = 1, \ldots, M)$:

$$R_Z^l : F_1^l \times \cdots \times F_p^l \to G^l = A^l \to G^l \tag{3.6}$$

R_Z^l is described by the MF $\mu_{R_Z^l}(\mathbf{x}, y)$, where

$$\mu_{R_Z^l}(\mathbf{x}, y) = \mu_{A^l \to G^l}(\mathbf{x}, y) \tag{3.7}$$

Consequently, when Mamdani implications are used, and multiple antecedents are connected by and [by t-norms, as in (2.117)], then

$$\begin{aligned} \mu_{A^l \to G^l}(\mathbf{x}, y) &= \mu_{F_1^l \times \cdots \times F_p^l \to G^l}(\mathbf{x}, y) = \mu_{F_1^l \times \cdots \times F_p^l}(\mathbf{x}) \bigstar \mu_{G^l}(y) = \mu_{F_1^l}(x_1) \bigstar \cdots \bigstar \mu_{F_p^l}(x_p) \bigstar \mu_{G^l}(y) \\ &= \left[T_{i=1}^p \mu_{F_i^l}(x_i) \right] \bigstar \mu_{G^l}(y) \end{aligned} \tag{3.8}$$

In (3.8), T is short for a t-norm.

The p-dimensional input to R_Z^l is given by the fuzzy set $A_{\mathbf{x}'}$ whose MF is [see (2.117)]

$$\mu_{A_{\mathbf{x}'}}(\mathbf{x}) = \mu_{X_1}(x_1 | x_1') \bigstar \cdots \bigstar \ \mu_{X_p}(x_p | x_p') = T_{i=1}^p \ \mu_{X_i}(x_i | x_i') \tag{3.9}$$

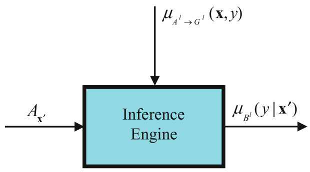


Fig. 3.2 Interpretation of fuzzy inference engine as a system

Each rule R_Z^l determines a fuzzy set $B^l = A_{\mathbf{x}'} \circ R_Z^l$ in Y such that [see (2.127)][5]

$$\mu_{B^l}(y|\mathbf{x}') = \mu_{A_{\mathbf{x}'} \circ R_Z^l}(y|\mathbf{x}') = \sup_{\mathbf{x} \in \mathbf{X}} \left[\mu_{A_{\mathbf{x}'}}(\mathbf{x}) \bigstar \mu_{A^l \to G^l}(\mathbf{x}, y) \right], y \in Y \tag{3.10}$$

This equation is the input–output relationship in Fig. 3.1 between the fuzzy set that excites a one-rule inference (engine) and the fuzzy set at the output of that engine. This sup-star composition is a highly nonlinear mapping from the input vector $\mathbf{x}'$ into a scalar output fuzzy set $\mu_{B^l}(y|\mathbf{x}')$ $(y \in Y)$.

The fuzzy inference engine can be interpreted as a *system*, one that maps fuzzy sets into fuzzy sets by means of the sup-star composition in (3.10). This is depicted in Fig. 3.2.

Substituting (3.8) and (3.9) into (3.10), it follows that $(l = 1, \ldots, M)$:

$$\begin{aligned}
\mu_{B^l}(y|\mathbf{x}') &= \sup_{\mathbf{x} \in X} \left[\mu_{A_{\mathbf{x}'}}(\mathbf{x}) \bigstar \mu_{A^l \to G^l}(\mathbf{x}, y) \right] = \sup_{\mathbf{x} \in X} \left[T_{i=1}^p \mu_{X_i}(x_i|x_i') \bigstar \left[T_{i=1}^p \mu_{F_i^l}(x_i) \right] \bigstar \mu_{G^l}(y) \right] \\
&= \sup_{\mathbf{x} \in X} \left\{ \left[T_{i=1}^p \mu_{X_i}(x_i|x_i') \bigstar \mu_{F_i^l}(x_i) \right] \bigstar \mu_{G^l}(y) \right\} \\
&= \left\{ \left[\sup_{x_1 \in X_1} \mu_{X_1}(x_1|x_1') \bigstar \mu_{F_1^l}(x_1) \right] \bigstar \cdots \bigstar \left[\sup_{x_p \in X_p} \mu_{X_p}(x_p|x_p') \bigstar \mu_{F_p^l}(x_p) \right] \right\} \bigstar \mu_{G^l}(y), \quad y \in Y
\end{aligned} \tag{3.11}$$

This is a very important sequence of calculations and result. The last line follows from the fact that $\mu_{X_i}(x_i|x_i') \bigstar \mu_{F_i^l}(x_i)$ is only a function of x_i, so that each supremum in (3.11) is over just a scalar variable, x_i .

A close examination of (3.11) reveals that, within the brace, the bracketed terms only involve interactions between a fuzzified input and its respective antecedent. It is only after all of those interactions have been computed that the braced term is t-normed with the entire consequent MF. In this book, the braced term is called the *firing level* for R_Z^l, and is denoted as $f^l(\mathbf{x}')$, where $f^l(\mathbf{x}') \in [0, 1]$, i.e.,

[5] Mendel (2001) did not use a conditioning notation in μ_{B^l}. It is now felt, by this author, that it is better to use such a conditioning notation, especially since the supremum operation in (3.10) is over $\mathbf{x} \in \mathbf{X}$, whereas the input to the fuzzy system occurs at a specific value of $\mathbf{x}$, namely $\mathbf{x} = \mathbf{x}'$. The notation $\mu_{B^l}(y|\mathbf{x}')$ makes it clear that μ_{B^l} changes with each $\mathbf{x}'$.

$$f^l(\mathbf{x}') \equiv \left\{ \left[\sup_{x_1 \in X_1} \mu_{X_1}(x_1|x') \bigstar \mu_{F_1^l}(x_1) \right] \bigstar \cdots \bigstar \left[\sup_{x_p \in X_p} \mu_{X_p}(x_p|x'_p) \bigstar \mu_{F_p^l}(x_p) \right] \right\} \tag{3.12}$$

The reader may find the notation $f^l(\mathbf{x}')$ confusing, because once the computations on the right-hand side of (3.12) are completed, then $f^l(\mathbf{x}')$ is a number. However, that number depends on $\mathbf{x}'$ through the way in which each $\mu_{X_i}(x_i|x'_i)$ is modeled (singleton or non-singleton fuzzifier, as is explained in Sect. 3.4.2), and so to show this dependency $f^l(\mathbf{x}')$ is used and not f^l .

Examining (3.12), observe that the interaction of each input with its antecedent contributes toward the rule's firing level. Denoting each input-antecedent interaction as $f^l(x'_i)$, where $f^l(x'_i) \in [0, 1]$, and

$$f^l(x'_i) \equiv \sup_{x_i \in X_i} \mu_{X_i}(x_i|x') \bigstar \mu_{F_i^l}(x_i), \tag{3.13}$$

it follows, from (3.12) and (3.13), that

$$f^l(\mathbf{x}') = T_{i=1}^p f^l(x'_i) \tag{3.14}$$

Consequently, (3.11) can be expressed very simply, as $(l = 1, \ldots, M)$:

$$\mu_{B^l}(y|\mathbf{x}') = f^l(\mathbf{x}') \bigstar \mu_{G^l}(y), \quad y \in Y \tag{3.15}$$

In words, (3.15) says that the MF for a fired rule (i.e., for the *fired-rule output set*) is the t-norm of the rule's firing level and the *entire* consequent fuzzy set. Clearly, if the firing level is zero, a rule does not fire, and the MF for such an unfired rule will be zero for $y \in Y$. Observe that, for a Mamdani fuzzy system, $\mu_{B^l}(y|\mathbf{x}')$, the MF of the fired-rule output fuzzy set B^l, is a function of y, where $y \in Y$.

Focusing next on the TSK rules in (3.2) and a TSK fuzzy system, following the same chain of reasoning used to obtain (3.14), and using (3.3), it should be obvious that, for TSK rules, (3.15) becomes $(l = 1, \ldots, M)$:

$$\mu_{B^l}(y|\mathbf{x}') = \mu_{B^l}(\mathbf{x}') = f^l(\mathbf{x}') \text{ when } y = g^l(\mathbf{x}') \tag{3.16}$$

Observe that, for a TSK fuzzy system, $\mu_{B^l}(\mathbf{x}')$ is an explicit function of $\mathbf{x}'$ and an implicit function of y, where $\mathbf{x}' \in X_1 \times X_2 \times \cdots \times X_p$; it is a spike of height $f^l(\mathbf{x}')$ at $y = g^l(\mathbf{x}')$. For those who are wondering at this point where or when the consequent $g^l(\mathbf{x}')$ will appear explicitly in the TSK fuzzy system, it will appear during the defuzzification step (see Sect. 3.6.4).

To summarize:

Theorem 3.1 *Let* $\mathbf{x}'$ *denote the input to a fuzzy system. Then the MF for a fired-rule output set,* B^l, is $(l = 1, \ldots, M)$:

$$\begin{cases} \text{Mamdani fuzzy system:} & \mu_{B^l}(y|\mathbf{x}') = f^l(\mathbf{x}') \bigstar \mu_{G^l}(y), \quad y \in Y \\ \text{TSK fuzzy system:} & \mu_{B^l}(\mathbf{x}') = f^l(\mathbf{x}') \text{ when } y = g^l(\mathbf{x}') \end{cases} \tag{3.17}$$

In (3.17), $f^l(\mathbf{x}')$ *is given by (3.12) [or (3.14) and (3.13)].*

3.4.2 Type-1 Rule Partitions

The firing levels partition $X_1 \times X_2 \times \cdots \times X_p$ in two interesting ways,[6] and since the formula for the firing level is exactly the same for type-1 Mamdani and TSK fuzzy systems these partitions are the same for both kinds of fuzzy systems.

Definition 3.6 A *type-1 first-order rule partition*[7] of $X_1 \times X_2 \times \cdots \times X_p$ is a collection of nonoverlapping hyper-rectangles in each of which a fixed number of rules is fired, where that number is found by examining the antecedent MFs, simultaneously.

Crisp rules have (type-0) first-order partitions, but, because of their nonoverlapping MFs, only one rule is fired in each and every one of them.

Definition 3.7 A *type-1 second-order rule partition* of $X_1 \times X_2 \times \cdots \times X_p$ occurs when the MF of a fuzzy set that is associated with either x_1, or x_2, or $\ldots$, or x_p, changes its mathematical formula within a type-1 first-order rule partition.

Another way to state this is that for a type-1 fuzzy system, the use of its MFs to compute the firing level within a type-1 first-order rule partition may be *adaptive* to the locations within that partition.

Type-1 first-order and second-order rule partitions are consequences of first-order uncertainty partitions (Definition 1.2). However, although a type-1 first-order rule partition of $X_1 \times X_2 \times \cdots \times X_p$ always exists, a type-1 second-order rule partition of $X_1 \times X_2 \times \cdots \times X_p$ may not exist (see Exercises 3.4 and 3.5).

Example 3.3 Suppose rules have two antecedents, x_1 and x_2, and that each of them is described by the three linguistic terms, *low* (L), *moderate* (M) and *high* (H), so that there are $3 \times 3 = 9$ rules. For illustrative purposes, *low* and *high* are described by trapezoidal shoulder MFs and *moderate* is described by an interior trapezoidal MF, as depicted on Fig. 3.3. Observe that X_1 and X_2 are jointly partitioned into five

[6]These are not fuzzy partitions, also known as Ruspini partitions (Ruspini 1969). See footnote 5 in Chap. 2.

[7]Wang (1997) calls these "partitions," but does not distinguish between first- and second-order partitions, something that only occurred to the author during the writing of this book. The nonoverlapping rectangles can also be called *information granules* (see Definition 6.16).

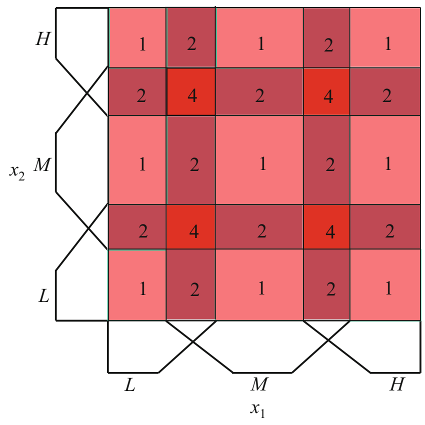


Fig. 3.3 Type-1 first-order rule partitions of $X_1 \times X_2$ for two-antecedent rules when each antecedent is described by the three linguistic terms, *low* (L), *moderate* (M), and *high* (H). The number in each *shaded rectangle* gives the number of rules that are fired in it

rectangles, three of which have no overlapping MFs and two of which have overlapping MFs. Consequently, $X_1 \times X_2$ has 25 *type-1 first-order rule partitions*. The numbers 1, 2, or 4 that appear in each of these type-1 first-order rule partitions denote how many two-antecedent rules are associated with that rectangle. For example,

- If numerical values of x_1 and x_2 both fall where their respective L has a flat top, then only one rule describes things, and the antecedent of that rule is

$$\text{IF } x_1 \text{ is } L \text{ and } x_2 \text{ is } L$$

- If numerical values of x_1 fall where its L has a flat top, but numerical values of x_2 fall where L and M overlap, then two rules describe things, and the antecedents of those rules are

$$\text{IF } x_1 \text{ is } L \text{ and } x_2 \text{ is } L$$

$$\text{IF } x_1 \text{ is } L \text{ and } x_2 \text{ is } M$$

- If numerical values of x_1 and x_2 fall where for each of them L and M overlap, then four rules describe things, and the antecedents of those rules are

$$\text{IF } x_1 \text{ is } L \text{ and } x_2 \text{ is } L$$

$$\text{IF } x_1 \text{ is } L \text{ and } x_2 \text{ is } M$$

$$\text{IF } x_1 \text{ is } M \text{ and } x_2 \text{ is } L$$

$$\text{IF } x_1 \text{ is } M \text{ and } x_2 \text{ is } M$$

Figure 3.4a shows the construction of a *type-1 second-order rule partition* diagram. Its additional dashed horizontal and vertical lines occur where a MF changes its mathematical formula in a type-1 first-order rule partition. It is these additional lines that further partition the type-1 first-order rule partitions into the type-1 second-order rule partitions.

Even when only one rule fires it is possible for the rule's firing level to adapt to its location (i.e., to change its numerical value) within a type-1 first-order rule partition. For example, refer to Fig. 3.4a to observe in its top row and first rectangle from the left that only one rule is activated, but different combinations of the MFs of the T1 FSs are used to compute the firing quantity. In fact, there are four type-1 second-order rule partitions, as the blow-up of this in Fig. 3.4b reveals. In each of these type-1 second-order rule partitions, the formulas for $\mu_L(x_1)$ and $\mu_H(x_2)$ are different, namely:

$$(\mu_H(x_2), \mu_L(x_1))_{11} = (1, 1)$$

$$(\mu_H(x_2), \mu_L(x_1))_{12} = (1, \mu_L(x_1))$$

$$(\mu_H(x_2), \mu_L(x_1))_{21} = (\mu_H(x_2), 1)$$

$$(\mu_H(x_2), \mu_L(x_1))_{22} = (\mu_H(x_2), \mu_L(x_1))$$

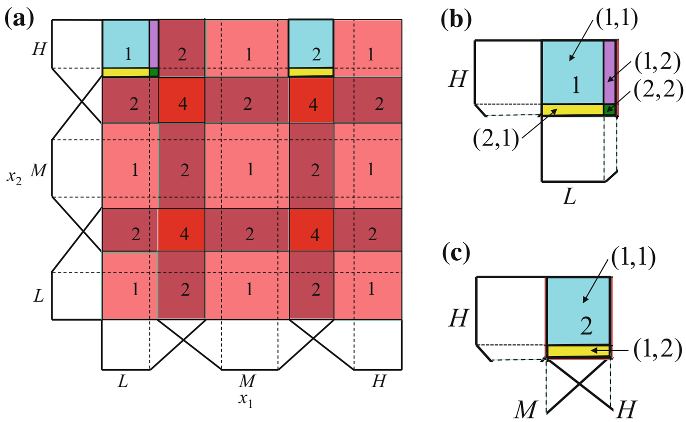


Fig. 3.4 **a** Construction of the type-1 second-order rule partition diagram of $X_1 \times X_2$ for Example 3.3 (the additional dashed horizontal and vertical lines occur where a MF changes it mathematical formula in a type-1 first-order rule partition); **b** blow-up of the Fig. 3.4(a) *top row* and first rectangle from the *left*; and, (c) blow-up of the Fig. 3.4(a) *top row* and fourth rectangle from the *left*. In (b) and (c) the MFs for x_1 and x_2 have been moved so that they are adjacent to the respective type-1 first-order rule partitions

As a second example, refer to Fig. 3.4a to observe in its top row and fourth rectangle from the left that two rules are activated, but again different combinations of the MFs of the T1 FSs are used to compute the firing quantity. In fact there are two type-1 second-order rule partitions, as the blow-up of this in Fig. 3.4c reveals. In each of these type-1 second-order rule partitions, the formulas for $\mu_L(x_1)$ and $\mu_H(x_2)$ are different, namely (now there are two rules):

$$(\mu_H(x_2), \mu_L(x_1))_{11} = (1, \mu_M(x_1)) \text{ and } (1, \mu_H(x_1))$$

$$(\mu_H(x_2), \mu_L(x_1))_{12} = (\mu_H(x_2), \mu_M(x_1)) \text{ and } (\mu_H(x_2), \mu_H(x_1))$$

As a third example, refer to any of the 4-rule rectangles in Fig. 3.4a to observe that they contain no type-1 second-order rule partitions.

It is straightforward to see that in the top row of Fig. 3.4a, there are $4 + 2 + 6 + 2 + 4 = 18$ type-1 second-order rule partitions; in the second row, there are $2 + 0 + 3 + 0 + 2 = 7$ type-1 second-order rule partitions, etc. The total number of type-1 second-order rule partitions is $18 + 7 + 27 + 7 + 18 = 77$. Interestingly, in this example it only takes nine rules to partition $X_1 \times X_2$ into 25 type-1 first-order rule partitions and 77 type-1 second-order rule partitions.

Be advised, that the results in this example depend very strongly on the shapes and overlap of the MFs. Exercises 3.3–3.6 ask the reader to repeat this example but for different kinds of MFs and overlap.

Example 3.4 If x_1 and x_2 are described by Q_1 and Q_2 terms, respectively, and the MFs of those terms look like the ones that are depicted in Fig. 3.3 for x_1 and x_2 (i.e., two shoulder MFs and $Q_i - 2$ interior trapezoidal MFs) then for each x_i there will be Q_i rectangles (partitions) of no overlapping MFs, and $Q_i - 1$ rectangles of two overlapping MFs. Consequently, $Q_1 \times Q_2$ rules will partition $X_1 \times X_2$ into $(2Q_1 - 1) \times (2Q_2 - 1)$ type-1 first-order rule partitions. In Example 3.3, $Q_1 = 3$ and $Q_2 = 3$ so that there are $5 \times 5 = 25$ type-1 first-order rule partitions.

The extensions of the results that are in Examples 3.3 and 3.4 to more than two inputs remain to be explored.

Whereas it is the nature of antecedent MFs that establish the type-1 first-and second-order rule partitions, it is the nature of each rule's consequent that fills each partition. Zadeh rules fill each type-1 rule partition with a linguistic term (or its surrogate, e.g., the COG of the MF for that term), whereas a TSK rule fills each rule partition with a mathematical function.

With reference to Fig. 3.3, a fuzzy system is said to be a *variable structure system*, i.e., for a given $\mathbf{x} = \mathbf{x}'$, only rules that are in a type-1 first-orderrule partition are activated, and those rules change as the system sweeps through its input space. This variable-structure property occurs automatically because of the way in which rules are created, through a partitioning of input variables into linguistic terms. Simply by changing the MFs or the number of linguistic terms, the structure of the variable-structure system can also be changed.

3.4.3 Fuzzification and Its Effects on Inference

In order to compute the firing level in (3.14), one needs to compute $f^l(x_i)$ in (3.13) $(i = 1, \ldots, p)$. This requires choosing a fuzzifier. In this section $f^l(x_i)$ is evaluated first for a singleton fuzzifier and then for a non-singleton fuzzifier.

3.4.3.1 Singleton Fuzzifier

For singleton fuzzification (Definition 3.4), the supremum operation in the sup-star composition (3.13) is very easy to evaluate because $\mu_{X_i}(x_i|x_i')$ is nonzero only at one point, $x_i = x_i'$; hence,

$$f^l(x_i') \equiv \sup_{x_i \in X_i} \mu_{X_i}(x_i|x_i') \bigstar \mu_{F_i^l}(x_i) = 1 \bigstar \mu_{F_1^l}(x_i') = \mu_{F_i^l}(x_i') \tag{3.18}$$

which is true for minimum or product t-norms. From (3.18) and (3.14), it follows that the firing level is

$$f^l(\mathbf{x}') = T_{i=1}^p f^l(x_i') = T_{i=1}^p \mu_{F_i^l}(x_i') \tag{3.19}$$

It is this tremendous simplification of the sup-star composition in (3.13), that is (in the opinion of this author), the reason for the popularity of singleton fuzzification; however, singleton fuzzification may not always be adequate, e.g., when data are corrupted by measurement noise. Non-singleton fuzzification provides a means for handling such uncertainties totally within the framework of a type-1 fuzzy system, and is the subject of Sect. 3.4.3.2.

Substituting (3.19) into (3.17), one has the following corollary to Theorem 3.1:

Corollary 3.1 *For* singleton fuzzification, *the MF for a* fired-rule output set, B^l, *is* $(l = 1, \ldots, M)$:

$$\begin{cases} \text{Mamdani fuzzy system:} & \mu_{B^l}(y|\mathbf{x}') = T_{i=1}^p \mu_{F_i^l}(x_i') \bigstar \mu_{G^l}(y), \quad y \in Y \\ \text{TSK fuzzy system:} & \mu_{B^l}(\mathbf{x}') = T_{i=1}^p \mu_{F_i^l}(x_i') \text{ when } y = g^l(\mathbf{x}') \end{cases} \tag{3.20}$$

Example 3.5 Here a pictorial description of (3.18)–(3.20) is obtained for a Mamdani fuzzy system that uses either minimum or product t-norms. This is done because fuzzy system designers are all familiar with such pictorial descriptions of a type-1 fuzzy system, since they provide them with a good understanding of some of the operations of such a system. It is also done because comparable pictorial descriptions for type-2 fuzzy systems are provided in later chapters, which can then be contrasted with the figures of this example to better understand the flow of uncertainties through type-2 fuzzy systems.

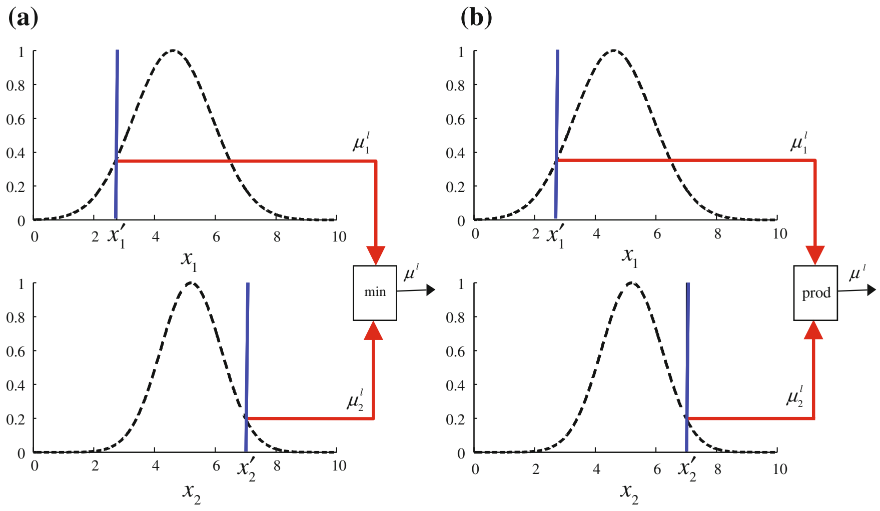


Fig. 3.5 Pictorial descriptions of input, antecedent operations, and firing level operations for a fuzzy system. Singleton fuzzification with **a** minimum t-norm, and **b** product t-norm. The *dashed curves* are the antecedent MFs

Figure 3.5 depicts input, antecedent and firing level computations [(3.18) and (3.19)] for a two-antecedent–single consequent rule, singleton fuzzification, and minimum or product t-norms. Note that μ_1^l and μ_2^l are short for $\mu_{F_1^l}(x_1')$ and $\mu_{F_2^l}(x_2')$, respectively; and, μ^l is short for $f^l(\mathbf{x}') = \mu_{F_1^l}(x_1')\bigstar\mu_{F_2^l}(x_2')$. In all cases, the firing level is a number equal to $\mu_{F_1^l}(x_1')\bigstar\mu_{F_2^l}(x_2')$. Observe, for example, that $\mu_{F_1^l}(x_1')$ occurs at the intersection of the vertical line at x_1' with $\mu_{F_1^l}(x_1)$.

For minimum t-norm $\mu_{F_1^l}(x_1')\bigstar\mu_{F_2^l}(x_2') = \min[\mu_{F_1^l}(x_1'), \mu_{F_2^l}(x_2')]$, whereas for product t-norm $\mu_{F_1^l}(x_1')\bigstar\mu_{F_2^l}(x_2') = \mu_{F_1^l}(x_1') \cdot \mu_{F_2^l}(x_2')$. The main thing to observe from these figures is that regardless of the t-norm, the result of input and antecedent operations is a number—the *firing level*.

Figure 3.6 depicts $\mu_{B^l}(y|\mathbf{x}') = \mu^l$ in the top line of (3.20) for a two-rule ($l = l_1, l_2$) Mamdani fuzzy system. It is obtained for each rule by t-norming the firing level term in (3.19) with $\mu_{G^l}(y)$ (the dashed curves) for $y \in Y$. Observe that $\mu_{B^l}(y|\mathbf{x}')$ for minimum t-norm is a clipped version of $\mu_{G^l}(y)$, where the clipping level equals the firing level for that rule. On the other hand, $\mu_{B^l}(y|\mathbf{x}')$ for product t-norm is a scaled version of $\mu_{G^l}(y)$, where the scaling level equals the firing level for that rule.

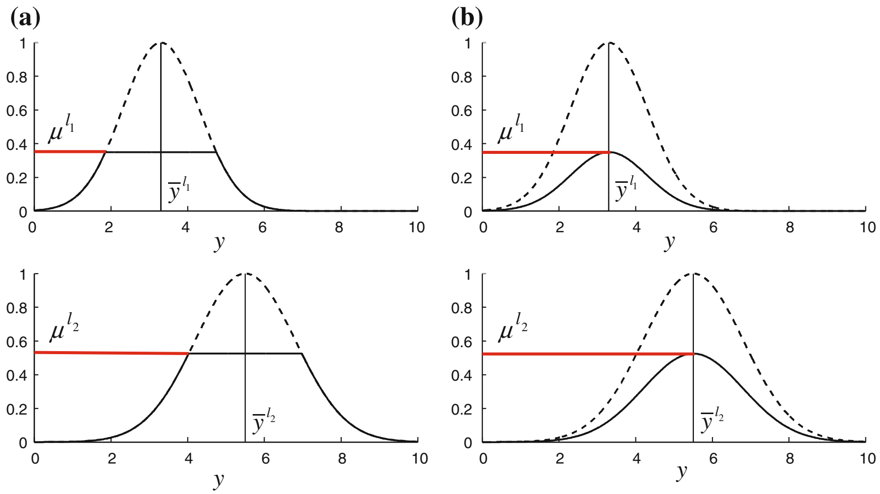


Fig. 3.6 Pictorial descriptions of consequent operations for a two-rule fuzzy system, $\mu_{B^l}(y|x'_1, x'_2)$. Fired-rule output sets with **a** minimum t-norm, and **b** product t-norm. The *dashed curves* are the consequent MFs

3.4.3.2 Non-Singleton Fuzzifier[8]

For non-singleton fuzzification (Definition 3.5),

$$f^l(x'_i) = \sup_{x_i \in X_i} \mu_{X_i}(x_i|x'_i) \bigstar \mu_{F_i^l}(x_i) \equiv \sup_{x_i \in X_i} \mu_{Q_i^l}(x_i|x'_i) \tag{3.21}$$

where[9]

$$\mu_{Q_i^l}(x_i|x'_i) \equiv \mu_{X_i}(x_i|x'_i) \bigstar \mu_{F_i^l}(x_i) \tag{3.22}$$

Unfortunately, the supremum operation in (3.21) is usually not easy to evaluate because $\mu_{X_i}(x_i|x'_i)$ is no longer nonzero only at one point. Examining (3.21), observe that it requires three computations: (1) $\mu_{Q_i^l}(x_i|x'_i)$, (2) $\arg\sup_{x_i \in X_i} \mu_{Q_i^l}(x_i|x'_i) \equiv x^l_{i,\max}$

[8]Readers not interested in non-singleton fuzzification can immediately go to Sect. 3.5.

[9]When the input is modeled as a type-1 fuzzy set, then a possible alternative to computing the firing level by means of the sup-star composition in (3.21) and (3.22) is to compute a similarity between X_i and F_i^l see, e.g., Raha, et al. (2002, 2008), Tsang, et al. (1995), Türksen and Zhong (1988), and Wagner, et al. (2016). When the Jaccard similarity measure is used, then this can be done using the results that are in Exercise 2.44. This approach remains to be explored when it is used in a real-time real-world application. Computing the similarity between a fuzzy input and its antecedent fuzzy set has found applicability in Computing With Words when the system inputs are restricted to a predetermined and known vocabulary of words that are modeled as type-1 fuzzy sets, because all of the similarity computations only need to be performed once ahead of time and can then be stored, as in Wu and Mendel (2009), Mendel and Wu (2010).

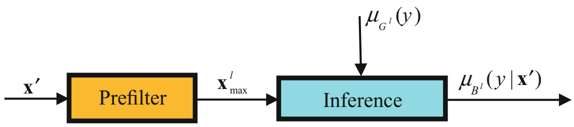


Fig. 3.7 Pre-filtering of the input to a non-singleton Mamdani fuzzy system

[where $x^l_{i,\max}$ is the value of x_i at which $\sup_{x_i \in X_i} \mu_{Q_i^l}(x_i|x_i')$ occurs], and (3) $f^l(x_i') = \mu_{Q_i^l}(x^l_{i,\max}|x_i')$. Consequently, for non-singleton fuzzification, (3.12) becomes

$$f^l(\mathbf{x}') = T^p_{i=1} f^l(x_i') = T^p_{i=1} \mu_{Q_i^l}(x^l_{i,\max}|x_i') \tag{3.23}$$

Substituting (3.23) into (3.17), one has another corollary to Theorem 3.1:

Corollary 3.2 *For* non-singleton fuzzification , *the*[10] *MF for a* fired-rule output set, B^l, *is* $(l = 1, \ldots, M)$:

$$\begin{cases} \text{Mamdani fuzzy system:} & \mu_{B^l}(y|\mathbf{x}') = T^p_{i=1} \mu_{Q_i^l}(x^l_{i,\max}|x_i') \bigstar \mu_{G^l}(y), \quad y \in Y \\ \text{TSK fuzzy system:} & \mu_{B^l}(\mathbf{x}') = \mu_{Q_i^l}(x^l_{i,\max}|x') \text{ when } y = g^l(\mathbf{x}') \end{cases} \tag{3.24}$$

Comparing (3.24) and (3.20), observe that a non-singleton fuzzy system first *pre-filters* its input $\mathbf{x}'$ transforming it to $\mathbf{x}^l_{\max}$. Doing this (see Fig. 3.7) accounts for the effects of the input measurement uncertainty. This pre-filtering is a direct result of the sup-star composition and occurs naturally within the existing framework of a fuzzy system. Interestingly enough, no such pre-filtering occurs naturally within the framework of neural networks.

Example 3.6 Here, analogous to the first part of Example 3.5, a pictorial description of (3.24) is obtained for a Mamdani fuzzy system that uses either the minimum or product t-norms. Figure 3.8 depicts Gaussian MFs (type-1 fuzzy numbers) that are centered about x_1' and x_2' . It also depicts $\mu_{Q_1^l}(x_1|x_1') \equiv \mu_{X_1}(x_1|x_1') \bigstar \mu_{F_1^l}(x_1)$ and $\mu_{Q_2^l}(x_2|x_2') \equiv \mu_{X_2}(x_2|x_2') \bigstar \mu_{F_2^l}(x_2)$ for a two-antecedent–single-consequent rule and minimum or product t-norms. The maximum values of $\mu_{Q_1^l}(x_1|x_1')$ and $\mu_{Q_2^l}(x_2|x_2')$

[10]In the first edition of this book (Mendel 2001) the following was stated "There does not seem to be any mention of a non-singleton TSK fuzzy logic system (FLS) in the literature; hence, this chapter focuses exclusively on singleton TSK FLSs—TSK FLSs, for short. Not being able to compensate for uncertain measurements, as we can do in a non-singleton Mamdani FLS, limits the applicability of TSK FLSs to situations where there either is no uncertainty (e.g., as in the design of deterministic TSK FL controllers) or all of the uncertainty can be accounted for just in the antecedent MFs." This statement no longer is true. Since a TSK FLS is very ad hoc, one can define the firing level any way that one wants to. As a result, the firing level for a TSK FLS can be defined for both singleton and non-singleton fuzzification, as has been done for the latter in the second line of (3.24).

are easy to visualize, and, when each maximum value is projected back down onto its respective x_1 or x_2 axis, $x^l_{1,\max}$ and $x^l_{2,\max}$ are located, respectively. On these figures, $\mu_{Q_1^l}(x^l_{1,\max}|x'_1)$ is called μ^l_1 and $\mu_{Q_2^l}(x^l_{2,\max}|x'_2)$ is called μ^l_2. Finally, $f^l(\mathbf{x}') \equiv \mu^l = \mu_{Q_1^l}(x^l_{1,\max}|x'_1) \bigstar \mu_{Q_2^l}(x^l_{2,\max}|x'_2)$ is computed in the min or prod blocks.

Figure 3.8 should be compared with Fig. 3.5. As in Fig. 3.5, the main thing to observe from Figs. 3.8a, b is that regardless of the t-norm, the result of input and antecedent operations is again a number—the *firing level*. Consequently, the results depicted in Fig. 3.6 [$\mu_{B^l}(y|\mathbf{x}')$ for a two-rule fuzzy system] remain the same for a non-singleton fuzzy system. The only difference between a non-singleton and singleton fuzzy system is the *numerical value* of the firing level. For a non-singleton fuzzy system this value includes the effects of input uncertainties, whereas for a singleton fuzzy system it does not.

Example 3.7 Here the results that were presented in Example 3.6 are quantified for the product t-norm. When all MFs are Gaussian, then it is straightforward to carry out the supremum computations of (3.21), as is demonstrated next.

The ith input fuzzy set and the corresponding rule antecedent fuzzy sets are described by the following MFs ($i = 1, \ldots, p$ and $l = 1, \ldots M$):

$$\mu_{X_i}(x_i|x'_i) = \exp\left\{-\tfrac{1}{2}[(x_i - x'_i)/\sigma_{X_i}]\right\}^2 \tag{3.25}$$

$$\mu_{F_i^l}(x_i) = \exp\left\{-\tfrac{1}{2}[(x_i - m_{F_i^l})/\sigma_{F_i^l}]\right\}^2 \tag{3.26}$$

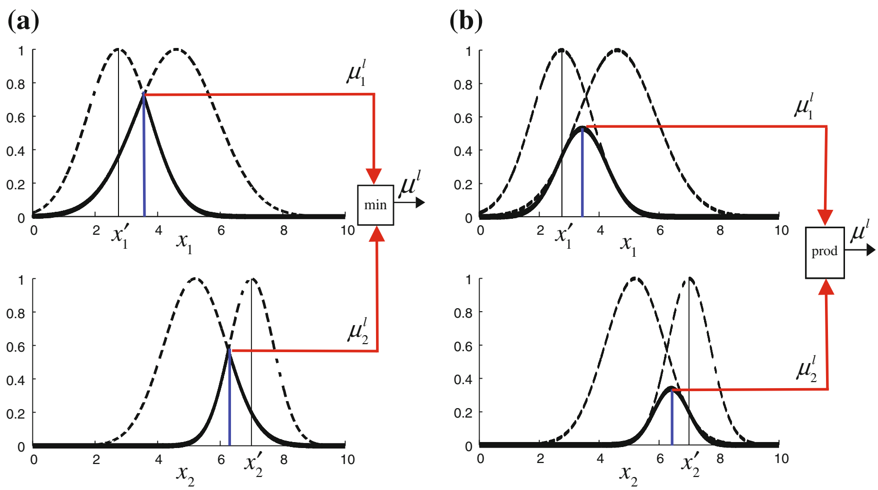


Fig. 3.8 Pictorial descriptions of input, antecedent operations, and firing-level operations for a fuzzy system. Non-singleton fuzzification with **a** minimum t-norm, and **b** product t-norm. The *dashed curves* are the input and antecedent MFs

Maximizing $\mu_{Q_i^l}(x_i|x_i')$, where

$$\mu_{Q_i^l}(x_i|x_i') = \mu_{X_i}(x_i|x_i')\mu_{F_i^l}(x_i) \tag{3.27}$$

one finds[11] (Example 2.34) ($i = 1, \ldots, p$ and $l = 1, \ldots, M$):

$$x_{i,\max}^l = (\sigma_{X_i}^2 m_{F_i^l} + \sigma_{F_i^l}^2 x_i')/(\sigma_{X_i}^2 + \sigma_{F_i^l}^2) \tag{3.28}$$

so that:

$$\mu_{Q_i^l}(x_{i,\max}^l|x_i') = \exp\{-(x_i' - m_{F_i^l})^2/2(\sigma_{X_i}^2 + \sigma_{F_i^l}^2)\} \equiv \mu_{Q_i^l}(x_i') \tag{3.29}$$

In the special but important case when all input points for each input variable have the same level of uncertainty (e.g., as in time-series forecasting), the spreads of the input sets will be the same, in which case $\sigma_{X_i}^2$ in (3.28) and (3.29) is a constant σ_X^2, so that these two equations become ($i = 1, \ldots, p$ and $l = 1, \ldots, M$):

$$x_{i,\max}^l = (\sigma_X^2 m_{F_i^l} + \sigma_{F_i^l}^2 x_i')/(\sigma_X^2 + \sigma_{F_i^l}^2) \tag{3.30}$$

$$\mu_{Q_i^l}(x_i') = \exp\{-(x_i' - m_{F_i^l})^2/2(\sigma_X^2 + \sigma_{F_i^l}^2)\} \tag{3.31}$$

The contribution to the firing level due to F_i^l, for singleton fuzzification, is $\mu_{F_i^l}(x_i')$, whereas for non-singleton fuzzification it is $\mu_{Q_i^l}(x_{i,\max}^l|x_i')$. Comparing (3.31) and (3.26) for when $x_i = x_i'$, observe that $\mu_{Q_i^l}(x_i') > \mu_{F_i^l}(x_i')$, which means that *the contribution to the firing level due to F_i^l is larger in the non-singleton case than it is in the singleton case.*

Note that when the uncertainty of the input becomes zero (i.e., $\sigma_X^2 = 0$), then (3.30) reduces to the singleton case, i.e., $x_{i,\max}^l = x_i'$, so that (3.25) and (3.27) become ($i = 1, \ldots, p$):

$$\mu_{X_i}(x_i = x_{i,\max}^l = x_i'|x_i') = \exp\left\{-\tfrac{1}{2}[(x_i' - x_i')/\sigma_{X_i}]\right\}^2 = 1 \tag{3.32}$$

$$\mu_{Q_i^l}(x_i|x_i') = \mu_{F_i^l}(x_i') \tag{3.33}$$

which agrees with (3.18).

[11]For minimum t-norm, $\mu_{Q_i^l}(x_i|x_i') = \min[\mu_{X_i}(x_i|x_i'), \mu_{F_i^l}(x_i)]$, and it is easy to show that when all MFs are Gaussian (Exercise 2.40) ($i = 1, \ldots, p$ and $l = 1, \ldots, M$): $x_{i,\max}^l = (\sigma_{X_i} m_{F_i^l} + \sigma_{F_i^l} x_i')/(\sigma_{X_i} + \sigma_{F_i^l})$.

Interestingly, (3.31) means that, in the special case of Gaussian MFs and product t-norm, it is possible to interpret the non-singleton fuzzy system as a singleton fuzzy system. The antecedent MFs of the latter are given by (3.31). Comparing (3.31) with (3.26), observe that the uncertainty about the measurements, contained in σ_X^2, causes the MFs in the equivalent singleton fuzzy system to become broader; i.e., $\sigma_{F_i^l}^2$ is replaced by $\sigma_{F_i^l}^2 + \sigma_X^2$. This broadening of the antecedent's MF causes more rules to fire in the non-singleton fuzzification case than in the singleton fuzzification case (i.e., the rectangles in a figure like Fig. 3.3, but for Gaussian MFs, become larger for two and four rules).

This example illustrates an important general fact: MF uncertainties can cause more rules to fire, as a hedge against such uncertainties.

3.5 Combining Fired-Rule Output Sets on the Way to Defuzzification

Observe, in Fig. 3.1, that the output(s) from the inference block go to the defuzzifier block; however, if more than one rule fires, a decision has to be made about what to do with them prior to defuzzification.

It is important to understand that in crisp logic only one rule can fire, because inputs can exactly match the antecedents of only one rule, and so the issue of combining multiply fired rules does not occur for a crisp system. Consequently, combining multiply fired rules causes a major road bump for a fuzzy system because it cannot be handled by extrapolating to a fuzzy system the way it was handled for a crisp system. Additionally there is no guidance about how to do this from the way a human combines rules (if indeed they do), because no one really knows.

As is usual for fuzzy sets and systems, there is no unique way to combine fired-rule output sets. In the rest of this section, three possibilities are described, the first two of which are only for a Mamdani fuzzy system, whereas the third is for both Mamdani and TSK fuzzy systems.

3.5.1 Mamdani Fuzzy System: Combining Using Set-Theoretic Operations

Zadeh (1973) combines Zadehrules using the connective word "else," one of whose definitions is *otherwise*. Lee (1990) uses the connective "also" and has a discussion about a number of studies that were performed to determine the best way to combine rules. Some people combine rules using a t—conorm—the fuzzy union—i.e., $B = B^1 \oplus \cdots \oplus B^M$ because this is very inclusive way of combining rules.

For a Mamdani fuzzy system, the final fuzzy set, B, is determined by all M rules, and can be obtained by combining B^l and its associated MF $\mu_{B^l}(y|\mathbf{x}')$ for all $l = 1, \ldots, M$, as:

$$B = A_{\mathbf{x}'} \circ [R_Z^1, \cdots, R_Z^M] \tag{3.34}$$

Lee (1990) provides a rigorous proof that the sup-min or sup-product compositions and connective "also," interpreted as a maximum t-conorm, are commutative, i.e.,

$$A_{\mathbf{x}'} \circ [R_Z^1, \cdots, R_Z^M] = \cup_{l=1}^M A_{\mathbf{x}'} \circ R_Z^l = \cup_{l=1}^M B^l \tag{3.35}$$

where the MF of B^l, $\mu_{B^l}(y|\mathbf{x}')$, is given in either the first line of (3.20) for singleton fuzzification, or in the first line of (3.24) for non-singleton fuzzification, so that *for a Mamdani fuzzy system*:

$$\begin{aligned} \mu_B(y|\mathbf{x}') &= \mu_{\cup_{l=1}^M B^l}(y|\mathbf{x}') \\ &= \begin{cases} \max\limits_{l=1,\ldots,M} T_{i=1}^p \mu_{F_i^l}(x_i') \bigstar \mu_{G^l}(y) \; y \in Y & \text{singleton fuzzification} \\ \max\limits_{l=1,\ldots,M} T_{i=1}^p \mu_{Q_i^l}(x_{i,\max}^l | x_i') \bigstar \mu_{G^l}(y) \; y \in Y & \text{non-singleton fuzzification} \end{cases} \end{aligned} \tag{3.36}$$

Instead of using the union, some prefer to use the intersection. Because the intersection is very exclusive (whereas the union is very inclusive), it has not found much popularity in the rule-based fuzzy systems literature and will not be discussed further here.

For additional discussions on connecting rules using set-theoretic operations, see: Dubois and Prade (1985), Kiska et al. (1985a, b, c), Lin and Lee (1996), Mizumoto (1987), Stachowicz and Kochanska (1987), Wang (1997), and Yen and Langari (1999).

Example 3.8 Figure 3.9 depicts the combined output set, $\mu_B(y|\mathbf{x}')$, for a two-rule fuzzy system, where the fired output sets are combined as in (3.36) using the maximum for the union. The results are fairly similar for both t-norms.

3.5.2 Mamdani Fuzzy System Combining Using a Weighted Combination

There does not appear to be a unique or compelling theoretical reason for combining rules using a set-theoretic operation. One alternative to doing this is to combine by forming an additive weighted combination of the MFs of the fired-rule output fuzzy sets as in Kosko (1992, 1997).

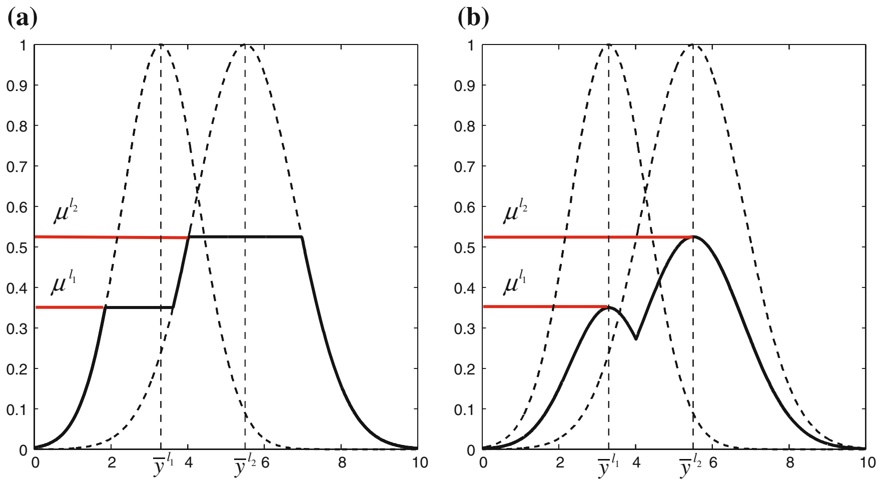


Fig. 3.9 Pictorial description of **a** combined output sets for the two fired output sets depicted in Fig. 3.6a, and **b** combined output sets for the two fired output sets depicted in Fig. 3.6b. The *dashed curves* are the consequent MFs

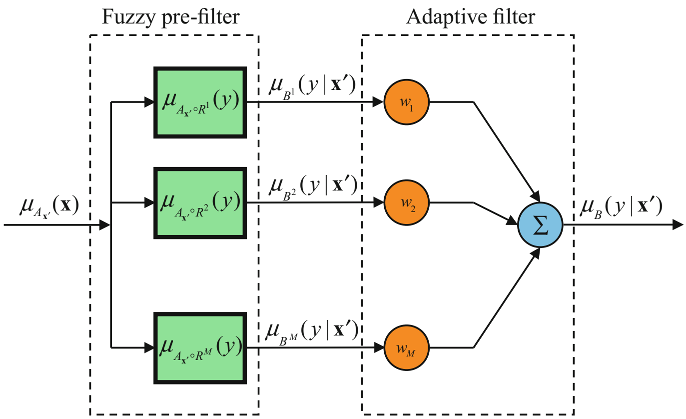


Fig. 3.10 An additive combiner that can be interpreted as an adaptive filter activated by fuzzy sets that are the output of a fuzzy pre-filter (Mendel 1995, © IEEE)

Figure 3.10 depicts an additive combiner. It resembles an adaptive filter whose inputs are the fired-rule output fuzzy sets. The weights of the combiner, $w_1, \ldots, w_M$, can be thought of as providing degrees of belief to each rule. It is conceivable that one knows that some rules are more reliable than others, in which case such rules would be assigned a larger weight than less reliable rules. If such information is not known ahead of time, then one either sets all the weights equal to unity or uses a training procedure to learn optimal values for the weights.

For a Mamdani fuzzy system,

$$\mu_B(y|\mathbf{x}') = \sum_{l=1}^{M} w_l \times f^l(\mathbf{x}') \bigstar \mu_{G^l}(y) \qquad y \in Y \tag{3.37}$$

where $w_l \in [0, 1]$, a constraint that is needed so that $\mu_B \in [0, 1]$, as is required by the MF of a fuzzy set. An easy way to achieve this constraint is to let $w_l \equiv w'_l / \sum_{l=1}^{M} w'$. In (3.37), $f^l(\mathbf{x}')$ is given in (3.19) for singleton fuzzification and in (3.23) for non-singleton fuzzification.

3.5.3 Mamdani and TSK Fuzzy Systems Combining During Defuzzification

In principle, one can combine in either of the two very different ways just described; however, when a fuzzy system is used in a real-world application, the issue of computational complexity becomes important, especially if the application is one that requires real-time processing.

For a Mamdani fuzzy system, taking the union of fired output sets or taking a linear combination of them requires additional computation time and storage, which many practitioners have found to be undesirable, and so an alternative to performing any kind of combining prior to defuzzification in a Mamdani fuzzy system is no combining, i.e., to go directly to defuzzification.

Direct defuzzification is always performed in a TSK fuzzy system.

Defuzzifications for Mamdani and TSK fuzzy systems are explained next in Sect. 3.6.

3.6 Defuzzifier

Definition 3.8 A *defuzzifier* produces a crisp output for the Fig. 3.1 fuzzy system from the combined (or not) fuzzy sets that appear at the output of its inference block. It is a mapping from one or more fuzzy sets into a real number.

Many defuzzifiers have been proposed in the literature for a Mamdani fuzzy system, but only two for a TSK fuzzy system. Because our interest is in real-world applications for a fuzzy system, one criterion for the choice of a defuzzifier is *computational simplicity*. This criterion has led to numerous candidates for defuzzifiers in a Mamdani fuzzy system, including: maximum, mean-of-maxima, centroid, center-of-sums, height, modified height, and center-of-sets.

The *maximum defuzzifier* examines the fuzzy set B in (3.36) and chooses as its output the value of y for which $\mu_B(y|\mathbf{x}')$ is a maximum. The *mean-of-maxima defuzzifier* examines this fuzzy set and first determines the values of y for which $\mu_B(y|\mathbf{x}')$ is a maximum, and then computes the mean of these values as its output. If the maximum value of $\mu_B(y|\mathbf{x}')$ only occurs at a single point, then the mean-of-maxima defuzzifier reduces to the maximum defuzzifier.

Both of these defuzzifiers can lead to peculiar results. For example, the maximum defuzzifier completely ignores the fact that $\mu_B(y|\mathbf{x}')$ is distributed (usually, non-symmetrically) over a range of y-values, and the mean-of-maxima defuzzifier may provide a zero mean value even though $\mu_B(y|\mathbf{x})$ is nonzero over most of its range. Even though these defuzzifers may be useful in some applications, they will not be used in this book.

A second criterion for the choice of a defuzzifier is *utilization in real-world applications*. The first edition of this book (Mendel 2001) explained five defuzzifiers for a Mamdani fuzzy system: centroid, center-of-sums, height, modified height, and center-of-sets (COS) defuzzifiers. In this edition, only the centroid, height, and center-of-sets defuzzifiers are explained (in Sects. 3.6.1–3.6.3) because, after more than 40 years of real-world applications of type-1 fuzzy systems, they are the ones that are the most used in such applications.

The two kinds of defuzzifiers for a TSK fuzzy system are described in Sect. 3.6.4.

3.6.1 Mamdani Fuzzy System: Centroid Defuzzifier

The centroid defuzzifier combines the type-1 fired-rule output fuzzy sets using union (i.e., a t-conorm, usually the maximum), as in (3.36), and then finds the centroid, $y_c(\mathbf{x}')$, of this set, i.e.,

$$y_c(\mathbf{x}') = \frac{\sum_{i=1}^{N} y_i \mu_B(y_i|\mathbf{x}')}{\sum_{i=1}^{N} \mu_B(y_i|\mathbf{x}')} \tag{3.38}$$

In (3.38), the MF for the output set B has been discretized at the N points, $y_1, \ldots, y_N$, and $y_c(\mathbf{x}')$ is shown as an explicit function of $\mathbf{x}'$ because $\mu_B(y_i|\mathbf{x}')$ is a function of fuzzy system input $\mathbf{x}'$, so, for each $\mathbf{x}'$ a different value is obtained for y_c.

The centroid defuzzifier, arguably the first kind of defuzzifier that was used in real-world applications of fuzzy systems, is usually difficult and time consuming to compute because of first having to compute the union in (3.36), and therefore it is not as widely used as it once was.

3.6.2 Mamdani Fuzzy System: Height Defuzzifier

The height defuzzifier (Driankov et al. 1996) (also called the center average defuzzifier (Wang 1994, 1997) replaces the fired-rule output set of each fired rule by a singleton at the point having maximum membership in the rule's consequent fuzzy set, with amplitude equal to the fired-rule output's MF at that point, and then calculates the centroid of the type-1 set comprised of these singletons. The output of a height defuzzifier, $y_h(\mathbf{x}')$, is given as

$$y_h(\mathbf{x}') = \frac{\sum_{l=1}^{M} \bar{y}^l \mu_{B^l}(\bar{y}^l|\mathbf{x}')}{\sum_{l=1}^{M} \mu_{B^l}(\bar{y}^l|\mathbf{x}')} \tag{3.39}$$

In (3.39), $\bar{y}^l$ is the point having maximum membership in the lth consequent fuzzy set (if there is more than one such point, their average can be taken as $\bar{y}^l$), and its membership grade in the lth fired-rule output set is $\mu_{B^l}(\bar{y}^l|\mathbf{x}')$. For singleton and non-singleton fuzzifications and a Mamdani fuzzy system, it follows from (3.20) and (3.24) that

$$\mu_{B^l}(\bar{y}^l|\mathbf{x}') = \begin{cases} \underbrace{\left[T_{i=1}^{p} \mu_{F_i^l}(x_i')\right]}_{f^l(\mathbf{x}')} \star \mu_{G^l}(\bar{y}^l) & \text{singleton fuzzification} \\ \underbrace{\left[T_{i=1}^{p} \mu_{Q_k^l}(x_{k,\max}^l|x_i')\right]}_{f^l(\mathbf{x}')} \star \mu_{G^l}(\bar{y}^l) & \text{non-singleton fuzzification} \end{cases} \tag{3.40}$$

Often $\mu_{G^l}(\bar{y}) = 1$.

Equation (3.39) is easily derived from calculus applied to the situation that is depicted in Fig. 3.11. Note, also, that although (3.39) and (3.38) look alike, they are quite different. Usually, $M \ll N$, so it is much easier to compute $y_h(\mathbf{x}')$ than it is to compute $y_c(\mathbf{x}')$.

A problem with the height defuzzifier occurs when a consequent set has its maximum value at $y = 0$ (e.g., this can occur for a left shoulder MF that is a negatively sloping line with maximum value at $y = 0$), in which case $\bar{y}^l = 0$. Examining (3.39), observe that such a rule never contributes anything to $y_h(\mathbf{x})$, which does not seem plausible. This problem is rectified by the center-of-sets defuzzifer.

3.6.3 Mamdani Fuzzy System: COS Defuzzifier

In center-of-sets (COS) defuzzification (Sugeno and Yasakuwa 1993, Karnik and Mendel 1998); each rule consequent set is replaced by a singleton located at its centroid, with amplitude equal to the firing level, after which the centroid of these

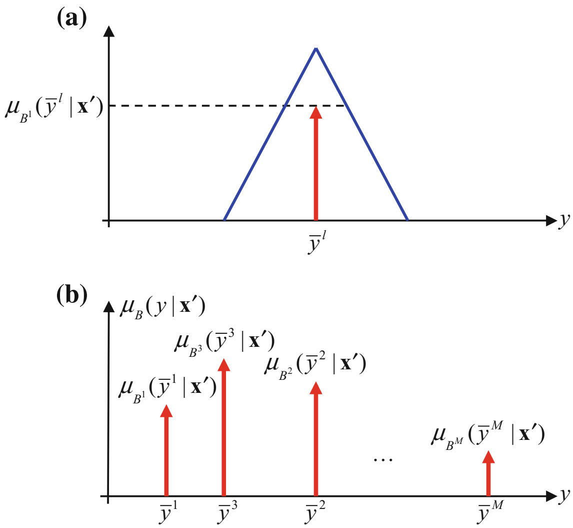


Fig. 3.11 **a** Spike of height $\mu_{B^l}(\bar{y}^l|\mathbf{x}')$ located at $\bar{y}^l$, and **b** discrete masses that are located at $\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^M$; each mass equals the peak value of an activated fired-rule output (rule) MF for R_Z^l, $\mu_{B^l}(\bar{y}^l|\mathbf{x}')$; an unactivated rule has zero value for $\mu_{B^l}(\bar{y}^l|\mathbf{x}')$, and the $\bar{y}^l$ do not have to appear in chronological order

singletons is found (see Fig. 3.12). The expression for the center-of-sets defuzzified output, $y_{COS}(\mathbf{x}')$, is given, as:

$$y_{\cos}(\mathbf{x}') = \frac{\sum_{l=1}^{M} \text{COG}(G^l) f^l(\mathbf{x}')}{\sum_{l=1}^{M} f^l(\mathbf{x}')} = \frac{\sum_{l=1}^{M} c^l f^l(\mathbf{x}')}{\sum_{l=1}^{M} f^l(\mathbf{x}')} \tag{3.41}$$

In (3.41), c^l is the centroid of the lth consequent set (for notational simplicity, c^l is used instead of c_{G^l}), and $f^l(\mathbf{x}')$ is the firing level that is given in (3.19) for singleton fuzzification, and in (3.23) for non-singleton fuzzification.

If each consequent is symmetric, normal, and convex, then $c^l = \bar{y}^l$, but for nonsymmetric consequent MFs $c^l \neq \bar{y}^l$. Additionally, for shoulder MFs c^l is never equal to $\bar{y}^l$ and so usually $y_{\text{COS}}(\mathbf{x}') \neq y_h(\mathbf{x}')$, although they may be numerically close.

The COS defuzzifier resolves the $\bar{y}^l = 0$ problem that was mentioned at the end of Sect. 3.6.2, because $c^l \neq 0$.

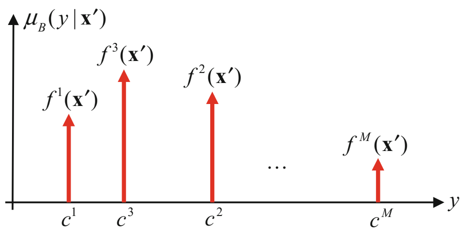


Fig. 3.12 Discrete masses are located at $c^1, c^2, \ldots, c^M$; each mass equals the firing level for R_Z^l; an unactivated rule has zero value for its firing level; and, the c^l do not have to appear in chronological order

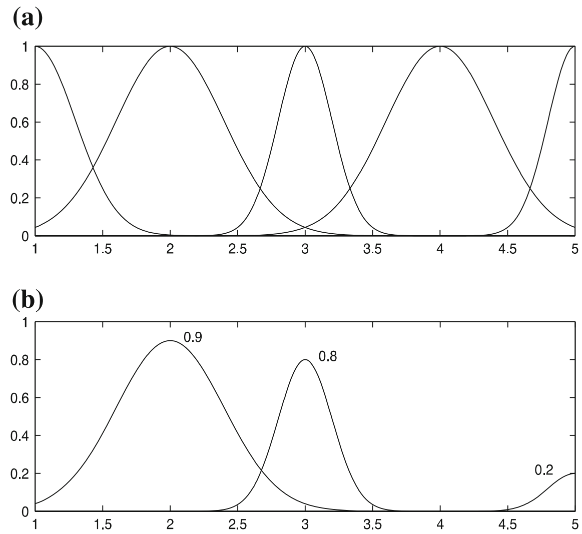


Fig. 3.13 **a** Consequent sets for Example 3.9. **b** Fired-rule output sets for some input $\mathbf{x} = \mathbf{x}'$

Example 3.9 Consider a five-rule Mamdani fuzzy system having consequent fuzzy sets that are depicted in Fig. 3.13a. Suppose that for some particular input $\mathbf{x} = \mathbf{x}'$ only three rules fire, and their fired-rule output sets are as depicted in Fig. 3.13b (assuming product inference). The numbers 0.9, 0.8, and 0.2 indicate the firing levels for each of the fired rules. The outputs of the three defuzzifiers (to two significant figures) for this example are: $y_c = 2.43$, $y_h = 2.74$ and $y_{\text{COS}} = 2.73$. Note that for the consequent set centered at 5, $\bar{y}^5 = 5$ but $c^5 = 4.84$; therefore, the outputs of the height and COS defuzzifiers for this example are slightly different.

3.6.4 TSK Fuzzy System Defuzzifiers

One kind of defuzzifier for a TSK fuzzy system leads to an *un-normalized TSK fuzzy system* as in Tanaka, et al. (1995), Tanaka and Sugeno (1998), for which

$$y_{TSK}^{U}(\mathbf{x}') = \sum_{l=1}^{M} f^l(\mathbf{x}')g^l(\mathbf{x}') \tag{3.42}$$

A second kind of defuzzifier leads to a *normalized TSK fuzzy system* for which

$$y_{TSK}^{N}(\mathbf{x}') = \frac{\sum_{l=1}^{M} f^l(\mathbf{x}')g^l(\mathbf{x}')}{\sum_{l=1}^{M} f^l(\mathbf{x}')} \tag{3.43}$$

In both (3.42) and (3.43), $f^l(\mathbf{x}')$ is the firing level that is given in (3.19) for singleton fuzzification, and in (3.23) for non-singleton fuzzification.

Observe that (3.43) has the same structure as $y_{\mathrm{COS}}(\mathbf{x}')$ in (3.41). It is only the function-consequent $g^l(\mathbf{x}')$ that makes (3.43) different from $y_{\mathrm{COS}}(\mathbf{x}')$.

When g^l is just a constant, i.e., $g^l(\mathbf{x}') = g^l$, then $y_{TSK}(\mathbf{x}')$ in (3.43) is structurally exactly the same as $y_{\mathrm{COS}}(\mathbf{x}')$ in (3.41). This is a very special kind of TSK fuzzy system, one that does not exploit the full potential of the more general consequent function $g^l(\mathbf{x})$. Although some people refer to this $y_{\mathrm{COS}}(\mathbf{x}')$ as a TSK fuzzy system, many people do not. The latter prefer to reserve the name "TSK fuzzy system" for the fuzzy system that uses the more general consequent function $g^l(\mathbf{x})$.

3.7 Comprehensive Example

Consider a fuzzy system with two inputs (x_1 and x_2) and one output (y). Each input domain consists of two shoulder type-1 fuzzy sets, shown in Fig. 3.14. The rule base consists of the following four Zadeh rules:

$$\begin{aligned}
R_Z^1 &: \text{IF } x_1 \text{ is } F_1^1 = X_{11} \text{ and } x_2 \text{ is } F_2^1 = X_{21}, \text{ THEN } y \text{ is } G^1 \\
R_Z^2 &: \text{IF } x_1 \text{ is } F_1^2 = X_{11} \text{ and } x_2 \text{ is } F_2^2 = X_{22}, \text{ THEN } y \text{ is } G^2 \\
R_Z^3 &: \text{IF } x_1 \text{ is } F_1^3 = X_{12} \text{ and } x_2 \text{ is } F_2^3 = X_{21}, \text{ THEN } y \text{ is } G^3 \\
R_Z^4 &: \text{IF } x_1 \text{ is } F_1^4 = X_{12} \text{ and } x_2 \text{ is } F_2^4 = X_{22}, \text{ THEN } y \text{ is } G^4
\end{aligned} \tag{3.44}$$

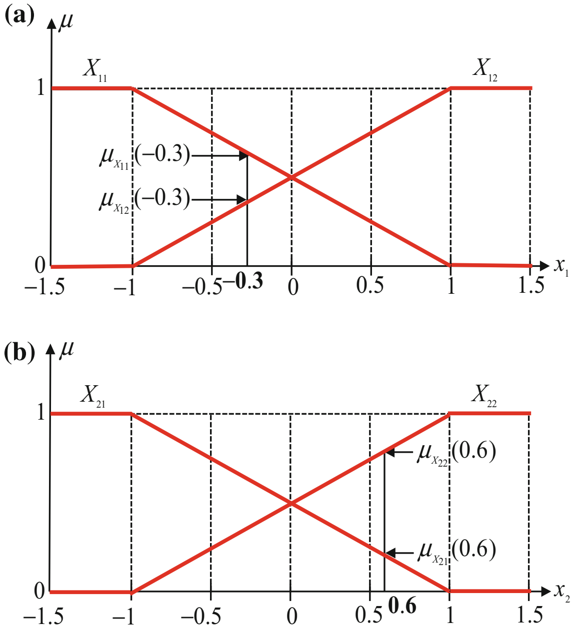


Fig. 3.14 MFs of the type-1 fuzzy sets, for: **a** x_1 and **b** x_2 . The measured values of $x_1(x_1' = -0.3)$ and x_2 $(x_2' = 0.6)$ are shown in bold face

Table 3.1 Rule-base and center of gravity (COG) of rule consequents of the fuzzy system

x_1	x_2	
	X_{21}	X_{22}
X_{11}	$R_Z^1 : \mathrm{COG}(G^1) \equiv -0.95$	$R_Z^2 : \mathrm{COG}(G^2) \equiv -0.5$
X_{12}	$R_Z^3 : \mathrm{COG}(G^3) \equiv 0.5$	$R_Z^4 : \mathrm{COG}(G^4) \equiv 0.95$

In (3.44), the antecedent fuzzy set for x_1 is the same in rules 1 and 2, namely X_{11}, and is also the same in rules 3 and 4, namely X_{12}, whereas the antecedent fuzzy set for x_2 is X_{21} for rules 1 and 3, and is X_{22} for rules 2 and 4.

For the purposes of this example, the center of gravity of the corresponding rule consequent was chosen a priori and is given in Table 3.1.

Here $\mathbf{x}' = col(-0.3, 0.6)$, whose components are shown bold-faced in Fig. 3.14. From that figure (or by writing the equations for the sloped portions of the four MFs and then evaluating them at $\mathbf{x}' = col(-0.3, 0.6)$, the following firing levels are obtained for each of the type-1 fuzzy sets:

$$\begin{cases} \mu_{X_{11}}(-0.3) = 0.65 \\ \mu_{X_{12}}(-0.3) = 0.35 \\ \mu_{X_{21}}(0.6) = 0.20 \\ \mu_{X_{22}}(0.6) = 0.80 \end{cases} \tag{3.45}$$

The firing levels for the four rules, computed using the product t-norm and singleton fuzzification, are shown in Table 3.2.

Using COS defuzzification, given by (3.41), and Table 3.2, one finds:

$$\begin{aligned} y_{\cos}(\mathbf{x}') &= \frac{f^1\mathrm{COG}(G^1) + f^2\mathrm{COG}(G^2) + f^3\mathrm{COG}(G^3) + f^4\mathrm{COG}(G^4)}{f^1 + f^2 + f^3 + f^4} \\ &= \frac{0.13 \times (-0.95) + 0.52 \times (-0.5) + 0.07 \times 0.5 + 0.28 \times 0.95}{0.13 + 0.52 + 0.07 + 0.28} \\ &= \frac{-0.124 - 0.26 + 0.035 + 0.266}{1} \\ &= -0.083 \end{aligned} \tag{3.46}$$

Table 3.2 Firing levels computed for $\mathbf{x}' = col(-0.3, 0.6)$ using (3.19), and rule consequent COG

Rule number	Firing level	Rule consequent (COG)
R_Z^1	$f^1 = \mu_{X_{11}}(-0.3) \times \mu_{X_{21}}(0.6) = 0.65 \times 0.20 = 0.13$	$\mathrm{COG}(G^1) \equiv -0.95$
R_Z^2	$f^2 = \mu_{X_{11}}(-0.3) \times \mu_{X_{22}}(0.6) = 0.65 \times 0.80 = 0.52$	$\mathrm{COG}(G^2) \equiv -0.5$
R_Z^3	$f^3 = \mu_{X_{12}}(-0.3) \times \mu_{X_{21}}(0.6) = 0.35 \times 0.20 = 0.07$	$\mathrm{COG}(G^3) \equiv 0.5$
R_Z^4	$f^4 = \mu_{X_{12}}(-0.3) \times \mu_{X_{22}}(0.6) = 0.35 \times 0.80 = 0.28$	$\mathrm{COG}(G^4) \equiv 0.95$

Next, the Zadeh rules are changed to the following TSK rules:

$$\begin{aligned}
R^1_{TSK}:&\text{ IF } x_1 \text{ is } F^1_1 = X_{11} \text{ and } x_2 \text{ is } F^1_2 = X_{21},\ \text{THEN } y \text{ is } g^1(\mathbf{x}) = c^1_0 + c^1_1 x_1 + c^1_2 x_2\\
R^2_{TSK}:&\text{ IF } x_1 \text{ is } F^2_1 = X_{11} \text{ and } x_2 \text{ is } F^2_2 = X_{22},\ \text{THEN } y \text{ is } g^2(\mathbf{x}) = c^2_0 + c^2_1 x_1 + c^2_2 x_2\\
R^3_{TSK}:&\text{ IF } x_1 \text{ is } F^3_1 = X_{12} \text{ and } x_2 \text{ is } F^3_2 = X_{21},\ \text{THEN } y \text{ is } g^3(\mathbf{x}) = c^3_0 + c^3_1 x_1 + c^3_2 x_2\\
R^4_{TSK}:&\text{ IF } x_1 \text{ is } F^4_1 = X_{12} \text{ and } x_2 \text{ is } F^4_2 = X_{22},\ \text{THEN } y \text{ is } g^4(\mathbf{x}) = c^4_0 + c^4_1 x_1 + c^4_2 x_2
\end{aligned} \tag{3.47}$$

The firing levels for the four rules, as well as their consequents, are shown in Table 3.3 for $\mathbf{x}' = col(-0.3, 0.6)$ Until numerical values are provided for all 12 rule consequent coefficients, numerical value for $y^U_{TSK}(\mathbf{x}')$ and $y^N_{TSK}(\mathbf{x}')$ cannot be computed. In this example, it is assumed that $c^1_0 = 1$, $c^2_0 = 2$, $c^3_0 = 3$, $c^4_0 = 4$, $c^1_1 = 1.5$, $c^2_1 = 2.5$, $c^3_1 = 3.5$, $c^4_1 = 4.5$, $c^1_2 = 2$, $c^2_2 = 2.5$, $c^3_2 = 3$ and $c^4_2 = 3.5$. The numerical values for $g^i(\mathbf{x}')$ that are in Table 3.3 follow accordingly.

For the un-normalized TSK fuzzy system, $y^U_{TSK}(\mathbf{x}')$ in (3.42) is:

$$\begin{aligned}
y^U_{TSK}(\mathbf{x}') &= \sum_{i=1}^{4} f_i(\mathbf{x}') g^i(\mathbf{x}')\\
&= 0.13 \times 1.75 + 0.52 \times 2.75 + 0.07 \times 3.75 + 0.28 \times 4.75 = 3.25
\end{aligned} \tag{3.48}$$

For the normalized TSK fuzzy system, $y^N_{TSK}(\mathbf{x}')$ in (3.43) is:

$$y^N_{TSK}(\mathbf{x}') = \frac{\sum_{i=1}^{4} f_i(\mathbf{x}') g^i(\mathbf{x}')}{\sum_{i=1}^{4} f_i(\mathbf{x}')} = \frac{3.25}{0.13 + 0.52 + 0.07 + 0.28} = 3.25 \tag{3.49}$$

It is purely coincidental that $\sum_{i=1}^{4} f_i(\mathbf{x}') = 1$, so that $y^U_{TSK}(\mathbf{x}') = y^N_{TSK}(\mathbf{x}')$; for other values of $\mathbf{x}'$, $y^U_{TSK}(\mathbf{x}') \neq y^N_{TSK}(\mathbf{x}')$

There is no point in comparing $y_{TSK}(\mathbf{x}')$ and $y_{\mathrm{COS}}(\mathbf{x}')$, because in this example the rule consequent COGs were chosen arbitrarily for the Mamdani fuzzy system, and the rule consequent parameters were chosen arbitrarily for the TSK fuzzy system. In practical applications, all of these parameters would be optimized so as to achieve (e.g., minimize) one or more application-specific performance metrics. It is only then that it is meaningful to compare $y^U_{TSK}(\mathbf{x}')$ or $y^N_{TSK}(\mathbf{x}')$ with $y_{\mathrm{COS}}(\mathbf{x}')$.

Table 3.3 Firing levels computed for $\mathbf{x}' = col(-0.3, 0.6)$ using (3.19), and the TSK rule consequents

Rule number	Firing level	Rule consequent
R^1_{TSK}	$f^1 = \mu_{X_{11}}(-0.3) \times \mu_{X_{21}}(0.6) = 0.65 \times 0.20 = 0.13$	$g^1(\mathbf{x}') = c^1_0 - 0.3c^1_1 + 0.6c^1_2 = 1.75$
R^2_{TSK}	$f^2 = \mu_{X_{11}}(-0.3) \times \mu_{X_{22}}(0.6) = 0.65 \times 0.80 = 0.52$	$g^2(\mathbf{x}') = c^2_0 - 0.3c^2_1 + 0.6c^2_2 = 2.75$
R^3_{TSK}	$f^3 = \mu_{X_{12}}(-0.3) \times \mu_{X_{21}}(0.6) = 0.35 \times 0.20 = 0.07$	$g^3(\mathbf{x}') = c^3_0 - 0.3c^3_1 + 0.6c^3_2 = 3.75$
R^4_{TSK}	$f^4 = \mu_{X_{12}}(-0.3) \times \mu_{X_{22}}(0.6) = 0.35 \times 0.80 = 0.28$	$g^4(\mathbf{x}') = c^4_0 - 0.3c^4_1 + 0.6c^4_2 = 4.75$

3.8 Fuzzy Basis Functions

The pictorial interpretations that have been provided for the inference block of the fuzzy system, in Figs. 3.5, 3.6 and 3.8, are informative; however, they do not provide one with a complete description of the fuzzy system. For such a description, one needs a mathematical formula that maps a crisp input $\mathbf{x}'$ into a crisp output $y = f(\mathbf{x}')$. From Fig. 3.1, observe that such a formula can be obtained by following the signal $\mathbf{x} = \mathbf{x}'$ through the fuzzifier, where it is converted into the fuzzy set $A_{\mathbf{x}'}$, into the inference block, where it is converted into the fuzzy sets B^l ($l = 1, \ldots, M$), after which it may (or may not) be combined into B, and finally into the defuzzifier, where it is converted into $f(\mathbf{x}')$. To write such a formula many choices must be made (they are formalized in Sect. 4.1).

Example 3.10 For a *Mamdani fuzzy system*, when either singleton or non-singleton fuzzification, max-product composition, product implication and height or center-of-sets defuzzification are chosen, leaving the choice of (normal) MFs open, it is easy to show that

$$y(\mathbf{x}') = \begin{cases} f_s(\mathbf{x}') = \dfrac{\sum_{l=1}^{M} \lambda^l \prod_{i=1}^{p} \mu_{F_i^l}(x_i')}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i^l}(x_i')} & \text{singleton fuzzification} \\ f_{ns}(\mathbf{x}') = \dfrac{\sum_{l=1}^{M} \lambda^l \prod_{i=1}^{p} \mu_{Q_i^l}(x_{i,\max}^l | x_i')}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{Q_i^l}(x_{i,\max}^l | x_i')} & \text{non-singleton fuzzification} \end{cases} \tag{3.50}$$

where $\lambda^l = \bar{y}^l$ for height defuzzification, and $\lambda^l = c^l$ for center-of-sets defuzzification. Subscript "s" ("ns") on $f_s(\mathbf{x}')$ $[f_{ns}(\mathbf{x}')]$ is used to remind one that this is a singleton (non-singleton) fuzzy system.

For height defuzzification, $f_s(\mathbf{x}')[f_{ns}(\mathbf{x}')]$ in (3.50) was obtained by starting with (3.39) and substituting for $\mu_{B^l}(\bar{y}^l|\mathbf{x}')$ from (3.40), i.e.,

$$\mu_{B^l}(\bar{y}^l|\mathbf{x}') = \begin{cases} \left[\prod_{i=1}^{p} \mu_{F_i^l}(x_i')\right] \times \mu_{G^l}(\bar{y}^l) = \prod_{i=1}^{p} \mu_{F_i^l}(x_i') = f^l(\mathbf{x}') & \text{singleton fuzzification} \\ \left[\prod_{i=1}^{p} \mu_{Q_i^l}(x_{i,\max}^l|x_i')\right] \times \mu_{G^l}(\bar{y}^l) = \prod_{i=1}^{p} \mu_{Q_i^l}(x_{i,\max}^l|x_i') = f^l(\mathbf{x}') & \text{non-singleton fuzzification} \end{cases} \tag{3.51}$$

where, because MFs are normal, $\mu_{G^l}(\bar{y}^l) = 1$. That $\prod_{i=1}^{p} \mu_{F_i^l}(x_i') = f^l(\mathbf{x}')$ for singleton fuzzification follows from (3.19), and that $\prod_{i=1}^{p} \mu_{Q_i^l}(x_{i,\max}^l|x_i') = f^l(\mathbf{x}')$ for non-singleton fuzzification follows from (3.23).

For center-of-sets defuzzification, one begins with (3.41) and proceeds in a similar manner to obtain (3.50).

Example 3.11 For a *Mamdani fuzzy system*, when either singleton or non-singleton fuzzification, max–min composition, minimum implication, [12] and height or center-of-sets defuzzification are chosen, again leaving the choice of (normal) MFs open, it is easy to show (in the manner of Example 3.10) that

$$y(\mathbf{x}') = \begin{cases} f_s(\mathbf{x}') = \dfrac{\sum_{l=1}^{M} \lambda^l \min_{i=1,\ldots,p}\left\{\mu_{F_i^l}(x_i')\right\}}{\sum_{l=1}^{M} \min_{i=1,\ldots,p}\left\{\mu_{F_i^l}(x_i')\right\}} & \text{singleton fuzzification} \\ f_{ns}(\mathbf{x}') = \dfrac{\sum_{l=1}^{M} \lambda^l \min_{i=1,\ldots,p}\left\{\mu_{Q_i^l}(x_{i,\max}^l | x_i')\right\}}{\sum_{l=1}^{M} \min_{i=1,\ldots,p}\left\{\mu_{Q_i^l}(x_{i,\max}^l | x_i')\right\}} & \text{non-singleton fuzzification} \end{cases} \tag{3.52}$$

Example 3.12 For a *TSK fuzzy system* in which the consequent functions depend linearly upon the inputs (which is the most common choice for these functions), i.e., $(l = 1, \ldots, M)$

$$g^l(\mathbf{x}) = c_0^l + \sum_{j=1}^{p} c_j^l x_j \tag{3.53}$$

when either singleton or non-singleton fuzzification, max-product composition, product implication, and a normalizedTSK fuzzy system are chosen, again leaving the choice of (normal) MFs open, it is easy to show (Exercise 3.13) that

$$y_{TSK}^N(\mathbf{x}') = \begin{cases} f_{s,TSK}(\mathbf{x}') = \dfrac{\sum_{l=1}^{M}\left[c_0^l + \sum_{j=1}^{p} c_j^l x_j'\right] \prod_{i=1}^{p} \mu_{F_i^l}(x_i')}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i^l}(x_i')} & \text{singleton fuzzification} \\ f_{ns,TSK}(\mathbf{x}') = \dfrac{\sum_{l=1}^{M}\left[c_0^l + \sum_{j=1}^{p} c_j^l x_j'\right] \prod_{k=1}^{p} \mu_{Q_k^l}(x_{k,\max}^l | x_i')}{\sum_{l=1}^{M} \prod_{k=1}^{p} \mu_{Q_k^l}(x_{k,\max}^l | x_i')} & \text{non-singleton fuzzification} \end{cases} \tag{3.54}$$

Definition 3.9 The Mamdani fuzzy systems in (3.50) and (3.52) can be represented more generally as the following *fuzzy basis function expansion*[13]:

$$y(\mathbf{x}) = \sum_{l=1}^{M} \lambda^l \phi_l(\mathbf{x}) \tag{3.55}$$

where $\phi_l(\mathbf{x})$ is called a *fuzzy basis function* (FBF) (Wang and Mendel 1992a, b). The TSK fuzzy system in (3.54) can also be represented more generally as the fuzzy basis function expansion:

---

[12]Fixing the composition and implication also fixes the t-norm.

[13]The FBFs are shown as a function of $\mathbf{x}$ rather than of $\mathbf{x}'$ since they are valid for $\mathbf{x} \in \mathbf{X}$.

$$y_{TSK}(\mathbf{x}) = \sum_{l=1}^{M} \sum_{j=0}^{p} c_j^l \phi_j^l(\mathbf{x}) \tag{3.56}$$

where $\phi_i^l(\mathbf{x})$ are its FBFs.

Example 3.13 Referring to Example 3.10 and (3.50), the FBFs are $(l = 1, \ldots, M)$:

$$\phi_l(\mathbf{x}) = \begin{cases} \dfrac{\prod_{i=1}^{p} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i^l}(x_i)} & \text{singleton fuzzification} \\ \dfrac{\prod_{k=1}^{p} \mu_{Q_k^l}(x_{k,\max}^l | x_i)}{\sum_{l=1}^{M} \prod_{k=1}^{p} \mu_{Q_k^l}(x_{k,\max}^l | x_i)} & \text{non-singleton fuzzification} \end{cases} \tag{3.57}$$

Referring to Example 3.11 and (3.52), the FBFs are $(l = 1, \ldots, M)$:

$$\phi_l(\mathbf{x}) = \begin{cases} \dfrac{\min_{i=1,\ldots,p}\left\{\mu_{F_i^l}(x_i)\right\}}{\sum_{l=1}^{M} \min_{i=1,\ldots,p}\left\{\mu_{F_i^l}(x_i)\right\}} & \text{singleton fuzzification} \\ \dfrac{\min_{i=1,\ldots,p}\left\{\mu_{Q_k^l}(x_{k,\max}^l | x_i)\right\}}{\sum_{l=1}^{M} \min_{i=1,\ldots,p}\left\{\mu_{Q_k^l}(x_{k,\max}^l | x_i)\right\}} & \text{non-singleton fuzzification} \end{cases} \tag{3.58}$$

Referring to Example 3.12 and (3.54), the FBFs are $(j = 0, 1, \ldots, p;\ l = 1, \ldots, M;\ x_0 \equiv 1)$

$$\phi_j^l(\mathbf{x}) = \begin{cases} \dfrac{\prod_{i=1}^{p} x_j \times \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i^l}(x_i)} & \text{singleton fuzzification} \\ \dfrac{\prod_{i=1}^{p} x_j \times \mu_{Q_i^l}(x_{i,\max}^l | x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{Q_i^l}(x_{i,\max}^l | x_i)} & \text{non-singleton fuzzification} \end{cases} \tag{3.59}$$

Referring to each fuzzy system as a FBF expansion is very useful, because it places a fuzzy system into the more global perspective of function approximation. A Mamdani fuzzy system has one FBF per rule, whereas a TSK fuzzy system, whose consequents are linear functions of the inputs, has $p + 1$ FBFs for each rule. Within the context of function approximation, this makes a TSK fuzzysystem arguably more powerful than a Mamdani fuzzy system, since it is a FBF expansion with $M(p + 1)$ terms, whereas a Mamdani fuzzy system is a FBF expansion with only M terms.

It is important to remember that the FBFs in (3.57) and (3.58) are valid only for very specific choices made about the fuzzifier, composition, implication, and defuzzifier, and the FBFs in (3.59) are valid only for very specific choices made about the consequent function, fuzzifier, composition, implication, and normalization. Changing any of these choices, (3.57)–(3.59) are no longer valid; but, the interpretation of a Mamdani or TSK fuzzysystem as a FBF expansion still is valid (except for centroid defuzzification of a Mamdani fuzzysystem[14]). Formulas that are comparable to these three can be derived for many other possibilities.

Although the index l on a FBF seems to be associated with a rule number, i.e., $l = 1, \ldots, M$, each FBF is affected by all of the rules because of the denominator in $\phi_l(\mathbf{x})$ or $\phi_j^l(\mathbf{x})$; hence, it is only partially correct to say that a specific FBF is associated just with one specific rule. Of course, if a rule is added or removed, thereby increasing or decreasing M, then one FBF is added or removed from the FBF expansion in (3.55), whereas $p+1$ FBFs are added or removed from the FBF expansion in (3.56). It is in that sense that it is correct to say that a specific FBF is associated with a specific rule.

The relationships between FBFs and other basis functions have been extensively studied in Kim and Mendel (1995). They are more general than radial basis functions, generalized radial basis functions, and hyper-basis functions as in Poggio and Girosi (1990). For very special choices of their parameters and singleton fuzzification, they bear structural resemblance to generalized regression neural networks in Specht (1991) and Gaussian sum approximations in Alspach and Sorenson (1972). The latter two begin by assuming that the measured data are random and that an estimate is desired of another random quantity. This bears no resemblance to our starting point for a fuzzy system where no assumption about randomness has been made.

The denominators of all FBFs serve to normalize their numerators. The numerators of the FBFs in (3.57) are radially symmetric[15]; hence, one could also refer to those FBFs as normalized radial basis functions. Such basis functions were originally suggested in Moody and Darkin (1989) as a means for sharing information across radial basis functions. Tao (1993) compared normalized and un-normalized radial basis functions, and demonstrated, by means of examples, the superiority of the former over the latter. It is important to note that the FBFs in (3.57) are normalized not by abstraction, as in Moody and Darkin (1989), but rather by design of the overall fuzzy system.

Many mathematical basis functions are orthogonal; but, not all basis functions have to be. FBFs are not, nor are the radial basis functions that are widely used in neural networks (e.g., Haykin 1996).

Example 3.14 Mendel (1995) What do the FBFs in (3.57) look like for singleton fuzzification? In order to answer this question, two situations are considered here—equally spaced and unequally spaced Gaussian antecedent MFs. To visualize the FBFs on a two-dimensional plot, let dim $\mathbf{x} = p = 1$, so that $\phi_l(\mathbf{x}) = \phi_l(x)$. Additionally, let the number of rules $M = 5$ and the standard deviations for all Gaussian antecedent MFs equal 10. Consequently $(l = 1, \ldots, 5)$,

[14] If $y_c(\mathbf{x}')$ in Eq. (3.38) is interpreted as a FBF expansion, then it would have N FBFs, where N is related to the discretization of Y. Change the discretization and N changes. To this author, the number of FBFs should not depend on such an N.

[15] Of course, this depends upon the symmetry of the $\mu_{F_i^l}(x_i)$.

$$\mu_{F^l}(x) = \exp\left\{-\tfrac{1}{2}[(x - m_{F^l})/10]\right\}^2 \tag{3.60}$$

and [e.g., top line of (3.57)]

$$\phi_l(\mathbf{x}) = \frac{\exp\{-\frac{1}{2}[(x - m_{F^l})/10]^2\}}{\sum_{l=1}^{5} \exp\{-\frac{1}{2}[(x - m_{F^l})/10]^2\}} \tag{3.61}$$

In the equally spaced situation, $m_{F^1} \equiv 20$, $m_{F^2} \equiv 35$, $m_{F^3} \equiv 50$, $m_{F^4} \equiv 65$ and $m_{F^5} \equiv 80$. Figure 3.15a depicts the five FBFs. Observe that the three interior FBFs are radially symmetric, whereas the two exterior FBFs are sigmoidal. These FBFs seem to combine the advantages of radial basis functions, which are good at characterizing local properties and sigmoidal neural networks, which are good at characterizing global properties.

Lest one believe that radial symmetry must always occur for interior FBFs, consider next the nonequally spaced situation, where $m_{F^1} \equiv 20$, $m_{F^2} \equiv 25$, $m_{F^3} \equiv$

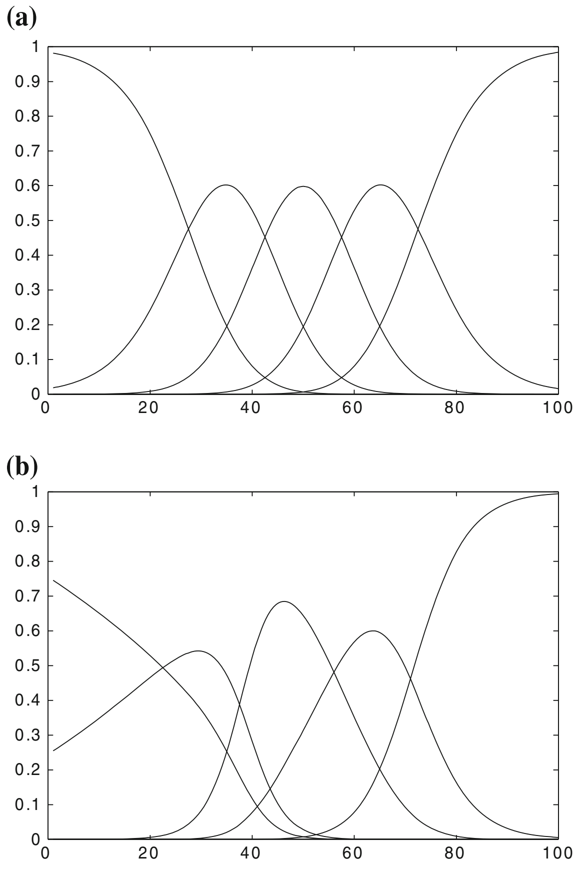


Fig. 3.15 Fuzzy basis functions for five rules in a singleton Mamdani fuzzy system **a** equally spaced, and **b** unequally spaced Gaussian antecedent MFs (Mendel 1995, © IEEE)

50, $m_{F^4} \equiv 62$ and $m_{F^5} \equiv 80$. Figure 3.15b depicts the five FBFs. Observe that the three interior FBFs are no longer radially symmetric, whereas the two exterior FBFs are still approximately sigmoidal. These figures should dispel the notion that fuzzy basis functions are radial basis functions. They are not; they are nonlinear functions of radially symmetrical functions [when the $\mu_{F_i^l}(x_i)$ are symmetrical].

Equally spaced FBFs are possible only when the mean values (centers) of the antecedent MFs are fixed by the designer. If these values are designed by means of a tuning procedure (as described in Sect. 4.2), so that they adapt to the data that is associated with the rules, then unequally spaced FBFs are the norm rather than the exception.

Example 3.15 Mendel (1995) This example parallels Example 3.14. What do the FBFs in (3.57) look like for non-singleton fuzzification? Again two situations are considered—equally spaced and unequally spaced Gaussian antecedent MFs and Gaussian fuzzy numbers for the fuzzified inputs. As in Example 3.14, $M = 5$, dim $\mathbf{x} = p = 1$, and standard deviations for all Gaussian antecedent MFs are set equal to 10. Additionally, standard deviations for the fuzzy input are set equal to 10. In this case, $\mu_{F^l}(x)$ are given by (3.60), and when the input is x',

$$\mu_X(x|x') = \exp\left\{-\tfrac{1}{2}[(x - x')/10]\right\}^2 \tag{3.62}$$

Using (3.31), it follows that $(l = 1, \ldots, 5)$:

$$\mu_{Q^l}(x') = \exp\left\{-\tfrac{1}{2}(x' - m_{F^l})^2/200\right\} \tag{3.63}$$

and therefore [e.g., the bottom line of (3.57)]

$$\phi_l(x) = \frac{\mu_{Q^l}(x)}{\sum_{l=1}^{5} \mu_{Q^l}(x)} = \frac{\exp\{-\frac{1}{2}(x - m_{F^l})^2/200\}}{\sum_{l=1}^{5} \exp\{-\frac{1}{2}(x - m_{F^l})^2/200\}} \tag{3.64}$$

In the equally spaced situation, as in Example 3.14, $m_{F^1} \equiv 20$, $m_{F^2} \equiv 35$, $m_{F^3} \equiv 50$, $m_{F^4} \equiv 65$ and $m_{F^5} \equiv 80$. Figure 3.16a depicts the five non-singleton FBFs as well as the five singleton FBFs [taken from Fig. 3.15a]. Observe that the FBFs for non-singleton fuzzification have longer tails and are broader than their singleton counterparts. This means that more of them will be activated in the non-singleton case than in the singleton case for a specific input value. For example, a vertical line at $x = x' = 30$ in Fig. 3.16a intersects three of the singleton FBFs and four of the non-singleton FBFs.

Input uncertainty tends to activate more FBFs (i.e., more rules are fired), which means (as already observed and mentioned just below the end of Example 3.7) that decisions are more distributed (across more rules) in the non-singleton case than in the singleton case.

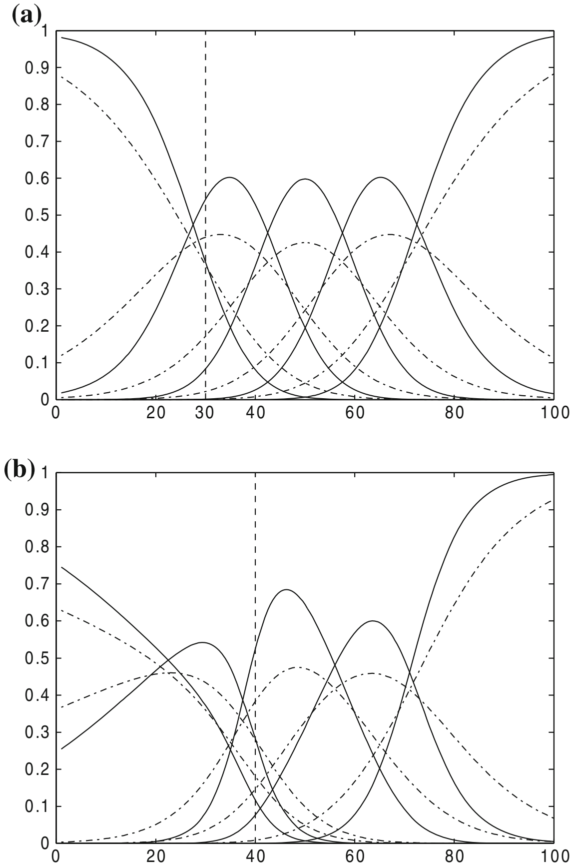


Fig. 3.16 Fuzzy basis functions for five rules in non-singleton (*dash-dotted lines*) and singleton (*solid lines*) Mamdani fuzzy systems. **a** Equally spaced and **b** unequally spaced Gaussian antecedent MFs and input Gaussian fuzzy numbers (Mendel 1995,© IEEE)

In the nonequally-spaced situation, $m_{F^1} \equiv 20$, $m_{F^2} \equiv 25$, $m_{F^3} \equiv 50$, $m_{F^4} \equiv 62$ and $m_{F^5} \equiv 80$. Figure 3.16b depicts the five non-singleton FBFs as well as the five singleton FBFs [which were plotted in Fig. 3.15b]. Note that a vertical line at $x = 40$ intersects four of the singleton FBFs and all five of the non-singleton FBFs. Even in the nonequally-spaced situation, input uncertainty tends to activate more FBFs.

Rules can come from numerical data or they can come from expert linguistic knowledge. Each rule contributes one basis function to a Mamdani FBF expansion.[16] Here the focus is only on the Mamdani FBF expansion and singleton fuzzification in (3.55). Extensions of the discussions below to other situations are left to the reader (Exercise 3.17).

---

[16]Expert rules have linguistic consequents and therefore are Zadeh rules. It is not clear that an expert can provide TSK rules due to their functional consequents.

It is convenient to decompose the right-hand side of (3.55) into the sum of two terms, one associated with FBFs that are associated with rules that come from numerical data and the other associated with rules that come from linguistic information, i.e.,

$$y(\mathbf{x}) = \sum_{l=1}^{M} \lambda^l \phi_l(\mathbf{x}) = f_{s,N}(\mathbf{x}) + f_{s,L}(\mathbf{x}) \tag{3.65}$$

in which $f_{s,N}(\mathbf{x})$ uses numerical-based fuzzy basis functions that are only normalized by all of the numerical-based fuzzy basis functions, and $f_{s,L}(\mathbf{x})$ uses linguistic-based fuzzy basis functions that are only normalized by all of the linguistic-based fuzzy basis functions. If one has a higher degree of belief in one set of rules over the other, then $f_{s,N}(\mathbf{x})$ and $f_{s,L}(\mathbf{x})$ can be combined in the following way:

$$y(\mathbf{x}) = \beta f_{s,N}(\mathbf{x}) + (1-\beta) f_{s,L}(\mathbf{x}) \quad 0 \le \beta \le 1 \tag{3.66}$$

When $\beta = 0$, then $y(\mathbf{x}) = f_{s,L}(\mathbf{x})$, which means, of course, that only linguistic information is used. On the other hand, if $\beta = 1$, then $y(\mathbf{x}) = f_{s,N}(\mathbf{x})$, which means that only numerical information is used. It is only when $0 < \beta < 1$ that both linguistic and numerical information are used.

This is not the only way that linguistic and numerical information can be combined. One deficiency in using (3.66) is that it does not produce a coupling between the linguistic and numerical FBFs. Such coupling occurs when the denominators of all the basis functions are made dependent on both linguistic and numerical information.

Example 3.16 How to achieve coupling between linguistic and numerical FBFs is illustrated here for singleton fuzzification, max-product composition, product implication, product t-norm, and COS defuzzification.

To begin rewrite (3.65) as

$$y(\mathbf{x}) = \sum_{l=1}^{M} \lambda^l \phi_l(\mathbf{x}) = \sum_{i=1}^{M_N} \lambda_N^i \phi_{N,i}(\mathbf{x}) + \sum_{j=1}^{M_L} \lambda_L^j \phi_{L,j}(\mathbf{x}) \tag{3.67}$$

where there are M_N FBFs associated with numerical data and M_L FBFs associated with linguistic information, and $M_N + M_L = M$. The FBFs are now given, from (3.57), as

$$\phi_{N,i}(\mathbf{x}) = \frac{\prod_{s=1}^{p} \mu_{F_s^i}(x_s)}{\sum_{l=1}^{M} \prod_{s=1}^{p} \mu_{F_s^l}(x_s)} \quad i = 1, \ldots, M_N \tag{3.68}$$

$$\phi_{L,j}(\mathbf{x}) = \frac{\prod_{s=1}^{p} \mu_{F_s^j}(x_s)}{\sum_{l=1}^{M} \prod_{s=1}^{p} \mu_{F_s^l}(x_s)} \quad j = 1, \ldots, M_L \tag{3.69}$$

Observe that the FBFs in (3.68) and (3.69) are normalized by information that is associated with both numerical and linguistic information, because their denominators depend on M, where $M = M_N + M_L$ hence, the numerical and linguistic basis functions are coupled through their denominators.

To date, FBFs are the only basis functions that can include linguistic information as well as numerical information; this makes them quite unique among all function approximation techniques.

3.9 Remarks and Insights

Before getting into the designs of fuzzy systems as well as some applications for them, there are many aspects of these systems that deserve some discussions. In this section, such discussions are provided about: layered architecture interpretations of a fuzzy system, universal approximation by fuzzy systems, continuity of fuzzy systems, rule explosion and some ways to control it, and rule interpretability.

3.9.1 Layered Architecture Interpretations of a Fuzzy System

By now, it should be very clear that there is a flow to the computations in Mamdani and TSK type-1 fuzzy systems, as is evidenced by the heavy-arrowed lines in Fig. 3.1. For example, for a Mamdani fuzzy system and non-singleton fuzzification ($i = 1, \ldots, p;\ l = 1, \ldots, M$):

- Inputs are fuzzified: $x'_i \to \mu_{X_i}(x_i|x'_i)$
- Firing levels are computed for each rule by the inference engine: $\left(\mu_{X_i}(x_i|x'_i), \mu_{F_i^l}(x_i)\right) \to f_i^l(x'_i) \to f^l(\mathbf{x}')$
- Fired-rule outputs may or may or not be combined (not shown on Fig. 3.1); if they are combined, then $\left(f^l(\mathbf{x}'), \mu_{G^l}(y)\right) \to \mu_{B^l}(y|\mathbf{x}') \to \mu_B(y|\mathbf{x}')$
- Combined fired-rule outputs sets are defuzzified into a crisp number: $\mu_B(y|\mathbf{x}') \to y_c(\mathbf{x}')$; or, if fired-rule output sets are not combined, then firing levels and, e.g., centroids of their respective consequent fuzzy sets are defuzzified into a crisp number: $(f^l(\mathbf{x}'), c^l) \to y_{\mathrm{COS}}(\mathbf{x}')$.

Some researchers attribute this flow of computations as a flow through a layered architecture or network. For example, Wang (1992a, (1997, p. 169) shows a three-layer network in which the first layer is where the inputs $\mathbf{x}'$ get fuzzified; the second layer is where firing levels are computed; and, the third layer is where

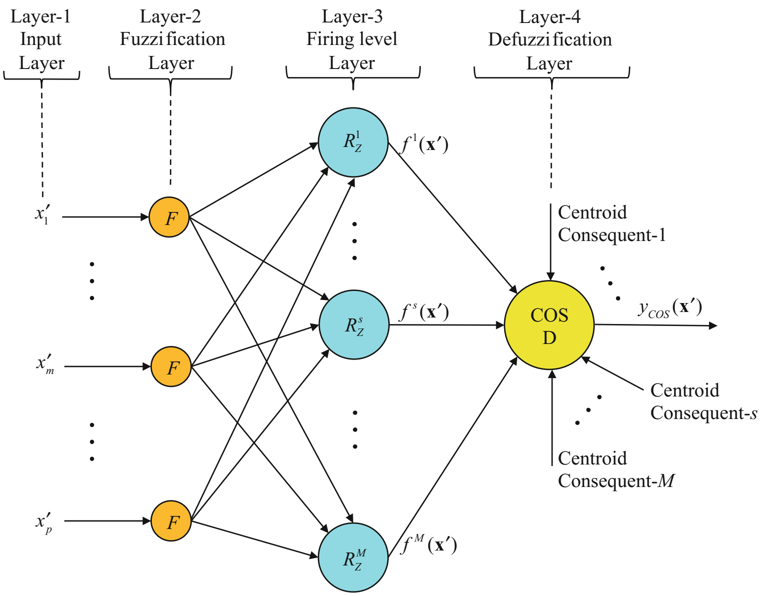


Fig. 3.17 Layered architecture summary of T1 Mamdani fuzzy system computations that use COS defuzzification

defuzzification occurs (fired-rule output sets are not combined in this figure). Other versions of this figure can be found in Lin and Lee (1991, 1996), Keller et al. (1992), and Jang (1993). Some of these figures show three layers, others show four or five layers.

Figure 3.17 is an example of one of these figures. Layer-2 is where the inputs are fuzzified (singleton, non-singleton). Layer-3 is where the firing level is computed for each of the M rules. Layer-4 is where defuzzification is performed and is illustrated for COS defuzzification, for which rule consequent centroids, which have been stored in memory, also have to be used.

When the antecedents of a type-1 fuzzy system include some time-delayed versions of the consequent, then such a type-1 fuzzy system is called *recurrent*. The earliest work on such a type-1 fuzzy system is Mouzouris and Mendel (1997); it used the word "dynamic" and not "recurrent". The layered architecture for a recurrent fuzzy system is similar to Fig. 3.17, but it includes feedback paths either in Layer 3 or between Layers 3 and 4.

A recurrent type-1 fuzzy system gives better results than the more usual (static) type-1 fuzzy system for identification of nonlinear dynamical systems. Additional references are Juang and Lin (1999), Zhang and Morris (1999), Lee and Teng, (2000), Lin and Chin (2004), Wang, et al. (2004), Juang (2002), Juang and Chen (2006) and Theocharis (2006).

Strangely (at least to this author), some of these layered architectures are given a "neural" designation, even though there is nothing neural about anything that has been done to derive a fuzzy system.[17] Arguably, this designation was used when it was recognized that the output of a fuzzy system could be expressed as a weighted combination of basis functions, and the computations could be organized in a layered manner, analogous to a neural network that has physical layers and weights. But this does not mean that a fuzzy system is a neural network. It is not!

3.9.2 Universal Approximation by Fuzzy Systems

How well does a type-1 fuzzy system approximate an unknown function? This is an important question that is asked about all types of function approximations, including the popular feedforward neural network (FFNN). Cybenko (1989), Hornik et al. (1989), Hornik (1993) as well as others e.g., Blum and Li (1991) demonstrated that a FFNN is a universal approximator, which means that a FFNN can uniformly approximate any real continuous function on a compact domain to arbitrary degree of accuracy. Hornik et al. (1989) used the Stone–Weirstrass theorem from real analysis to prove this result.

There is now a very large literature about different configurations of Mamdani fuzzy systems that are universal approximators e.g., Buckley (1992, 1993), Castro (1995), Kosko (1994), Kreinovich et al. (1998, 1999), Wang and Mendel (1992a, b) most of which are summarized very well in Kreinovich et al. (1998). Wang and Mendel (1992a) and Wang (1992b) also used the Stone–Weirstrass theorem to prove universal approximation for a Mamdani singleton type-1 fuzzy system that uses product composition, product implication, Gaussian MFs, and height defuzzification. Kosko (1992, 1994) proved a similar result for an additive fuzzy system, one that uses singleton fuzzification, centroid defuzzification, product composition, and product implication using the concept of *fuzzy patches*. Mouzouris and Mendel (1997) have a universal approximation theorem for non-singleton fuzzy systems. Kreinovich et al. (1999) further showed that Mamdani type-1 fuzzy systems are universal approximators not only for a smooth function but also for its derivatives. Castro (1995) showed that Mamdani type-1 fuzzy systems with Gaussian, triangular, or trapezoidal MFs, any t-norm, and any practical defuzzification method are universal approximators.

Just as Mamdani fuzzy systems have been proved to be universal approximators, so have TSK fuzzy systems [e.g., Buckley (1993), Tanaka and Wang (2001, Ch. 14)].

---

[17]As strange, is the transcription of the word "Network" in ANFIS (Jang 1993) to "Neural" by many others. Perhaps this is due to the statements that appear on pages 666 and 670 of this paper: "This section [III] introduces the architecture and learning procedure of the adaptive network which is in fact a superset of all kinds of feedforward neural networks with supervised learning capability. … In this section [IV], we propose a class of adaptive networks which are functionally equivalent to fuzzy inference systems."

A universal approximation theorem is an existence theorem. It helps to explain why a fuzzy system is so successful in engineering applications; however, it does not explain how to specify such a system. The same is true for a FFNN universal approximation theorem, since it does not indicate how many layers of neurons should be used, how many neurons should be used in each layer, or how interconnected the neurons should be. Universal approximation theorems for a FFNN imply that by using *enough* layers, *enough* neurons in each layer, and *enough* interconnectivity, the FFNN can uniformly approximate any real continuous nonlinear function to arbitrary degree of accuracy. For a fuzzy system, universal approximation theorems imply that by using enough terms for each input variable, and enough rules, the fuzzy system can also uniformly approximate any real continuous nonlinear function to arbitrary degree of accuracy.

How to actually specify and design a fuzzy system is the subject of Chap. 4.

3.9.3 Continuity of Fuzzy Systems

Wu and Mendel (2011) examine the continuity of the output of a type-1 fuzzy system. This is very technical and so only their findings will be stated here. They are for when: (1) $y(\mathbf{x})$ is computed as in (3.39), or as in (3.41), using product or minimum t-norm: (2) each component of $\mathbf{x}$ $[x_i\,(i=1,\ldots,p)]$ is described by Q_i type-1 fuzzy sets, $X_{i1},\ldots,X_{iQ_i}$; and, (3) there are no gaps[18] in the coverage of $\mu_{X_{ij}}(x_i)(j=1,\ldots,Q_i)$. Their findings are

- When $y(\mathbf{x})$ is a universal approximator of a continuous function[19] $g(\mathbf{x})$, then $y(\mathbf{x})$ is also a continuous function.
- $y(\mathbf{x})$ is continuous at $\mathbf{c}=(c_1,\ldots,c_p)$ if and only if $\max_{j=1,\ldots,Q_i}\mu_{X_{ij}}(c_i)>0$ for $i=1,\ldots,p$, i.e., every c_i is covered by some continuous type-1 fuzzy sets, which means that the membership grade of c_i on at least one of the fuzzy sets $X_{i1},\ldots,X_{iQ_i}$ is larger than 0.
- $y(\mathbf{x})$ has a *gap discontinuity*[20] at $\mathbf{c}=(c_1,\ldots,c_p)$ if and only if there exists a c_i such that $\max_{j=1,\ldots,Q_i}\mu_{X_{ij}}(c_i)=0$, which means that there is at least one c_i not covered by any continuous type-1 fuzzy set $X_{i1},\ldots,X_{iQ_i}$ in its domain.

---

[18] A gap in coverage occurs when at least one point in the input domain of x_i is not covered by the MFs of $X_{i1},\ldots,X_{iQ_i}$.

[19] As is stated in Wu and Mendel (2011, p. 180): The single-variable function $f(x)$ is *continuous at c* if and only if $f(x)$ is defined at c, and $\lim_{x\to c}f(x)=f(c)$, i.e., for any $\varepsilon>0$, there exists $\delta>0$ such that $|x-c|<\delta\Rightarrow|f(x)-f(c)|<\varepsilon$. Additionally, a multivariable function $f(\mathbf{x})$ is continuous at $\mathbf{c}=(c_1,\ldots,c_p)$ if and only if it is defined at $\mathbf{c}$, and for any $\varepsilon>0$, there exists $\delta>0$ such that $\max_{i=1,\ldots,p}|x_i-c_i|<\delta\Rightarrow|f(\mathbf{x})-f(\mathbf{c})|<\varepsilon$. A type-1 fuzzy set F is continuous if and only if $\mu_F(x)$ is a continuous function of x, $x\in X$.

[20] $y(\mathbf{x})$ has a *gap discontinuity* at $\mathbf{c}$ if $y(\mathbf{c})$ is undefined.

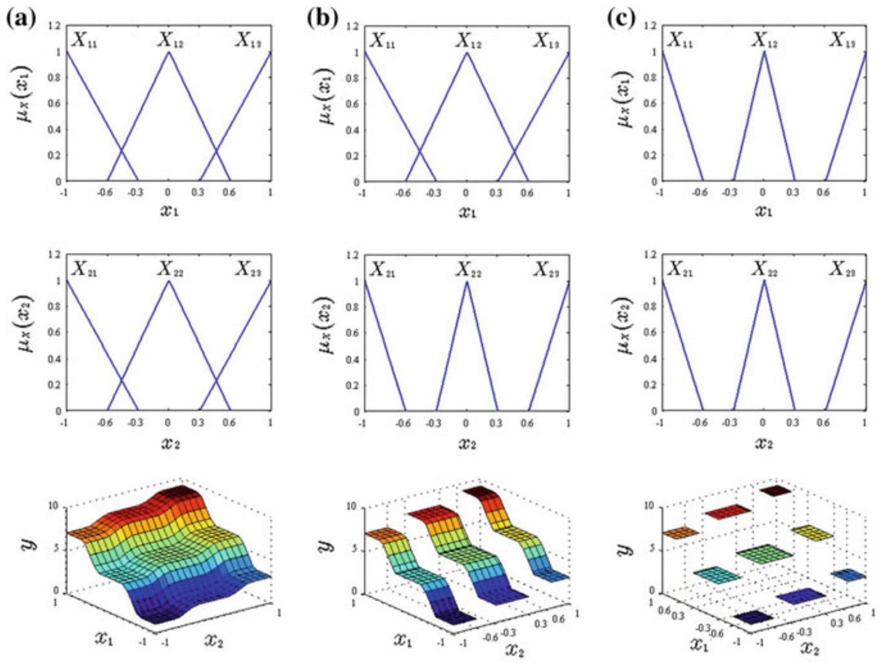


Fig. 3.18 Examples of input–output mappings $y_{\text{COS}}(x_1, x_2)$ for a type-1 fuzzy system that has two inputs. **a** Input MFs fully cover D_{X_1} and D_{X_2}; **b** Input MFs fully cover D_{X_1} but not D_{X_2}; and **c** Input MFs do not fully cover D_{X_2} and D_{X_2} (Wu and Mendel 2011, © 2011 IEEE)

Table 3.4 Rule-base for the type-1 fuzzy systems shown in Fig. 3.18 [Wu and Mendel (2011, Table III)]; the numbers are c^l

$x_1 \backslash x_2$	X_{21}	X_{22}	X_{23}
X_{11}	$R_Z^1 : 1$	$R_Z^2 : 2$	$R_Z^3 : 3$
X_{12}	$R_Z^4 : 4$	$R_Z^5 : 5$	$R_Z^6 : 6$
X_{13}	$R_Z^7 : 7$	$R_Z^8 : 8$	$R_Z^9 : 9$

Example 3.17 [Adapted from Wu and Mendel (2011, pp. 181–182)] The three columns of[21] Fig. 3.18 depict $y(x_1, x_2)$ versus x_1 and x_2 for three type-1 Mamdani singleton fuzzy systems that use center-of sets defuzzification and product t-norm, i.e. $y(x_1, x_2) = y_{\text{COS}}(x_1, x_2)$ computed by (3.41). Table 3.4 provides the nine rules R_Z^l $(l = 1, \ldots, 9)$ that were used for these fuzzy systems. Each entry in the table corresponds to the centroid of a rule's consequent, c^l, whose numerical value equals the rule number (l), which is only for illustrative purposes.

[21]The colorized Fig. 3.18 was provided by Dongrui Wu.

From Fig. 3.18, observe that

1. When the input MFs fully cover the input domains, as in Fig. 3.18a, $y_{\mathrm{COS}}(x_1, x_2)$ is continuous.
2. When at least one point in the input domain is not covered by the MFs, the corresponding $y_{\mathrm{COS}}(x_1, x_2)$ has gap discontinuities, as in Fig. 3.18b, c.
3. The gaps in the output domain are determined by the uncovered intervals in the input domains, e.g., in Fig. 3.18b, x_2 is uncovered at $[-0.6, -0.3] \cup [0.3, 0.6]$, and hence $y_{\mathrm{COS}}(x_1, x_2)$ has gap discontinuities at $[-0.6, -0.3] \cup [0.3, 0.6]$ only in the x_2 domain. Similarly, as shown in Fig. 3.18c, both x_1 and x_2 are uncovered at $[-0.6, -0.3] \cup [0.3, 0.6]$, and hence, $y_{\mathrm{COS}}(x_1, x_2)$ has gap discontinuities at $[-0.6, -0.3] \cup [0.3, 0.6]$ in both x_1 and x_2 domains.

In retrospect, the results found for continuity of type-1 fuzzy systems are not surprising. However, they demonstrate that when MF parameters are optimized (as explained in Sects. 4.1 and 4.2) and it is required that the output mapping of the fuzzy system must be continuous, then the MF parameters must be constrained so that MF gaps do not occur. When triangle or trapezoidal MFs are used, there can be continuity problems, and constraints have to be imposed on these MFs to avoid them. Interestingly, *when Gaussian MFs are used, gaps can never occur (because, in theory, such MFs cover their entire domains) and so continuity is not an issue*. This is arguably a pretty strong reason to use Gaussian MFs.

3.9.4 Rule Explosion and Some Ways to Control It

Rule explosion refers to the geometric increase in the number of rules that can occur either as the number of antecedents increases or as the number of MFs for each antecedent increase, and it can be a problem for a fuzzy system. Some design methods have been proposed to eliminate it. Wang and Mendel (1992a) proposed an orthogonal least squares method to select the most important fuzzy basis functions, each of which is associated with a specific rule. Yen and Wang (1999) used an eigenvalue decomposition method to construct a reduced fuzzy model. Mouzouris and Mendel (1996, 1997b) used a Singular Value Decomposition (SVD)–QR method to extract the most important fuzzy rules from a given rule base. Yen and Wang (1996) proposed using a direct SVD method to determine the number and positions of the most significant fuzzy rules. Yam et al. (1999) also used the SVD for rule reduction. Using SVD to accomplish rule reduction is described in Sect. 4.2.4.

Other approaches for controlling rule explosion are:

1. Fix the number of rules ahead of time (this is done in Sect. 4.3 for the application of time-series forecasting, in which the rule antecedents are time-delayed versions of one another).

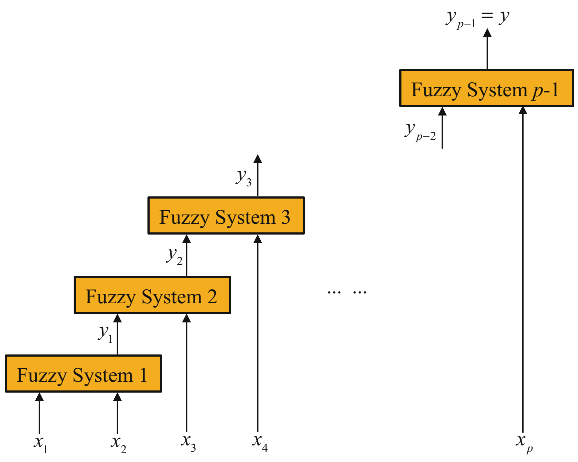


Fig. 3.19 An example of p-1 input hierarchical fuzzy system that is comprised of p-1 two-input fuzzy systems (Wang (1999, © 1999 IEEE)

2. Use rule interpolation when data produce a sparse set of rules, i.e., when rules do not cover all of $X_1 \times X_2 \times \cdots \times X_p$ [see, e.g., Kóczy and Hirota (1997), Wang et al. (2005), Kovács (2009)].
3. Map the Zadeh rules in (3.1), that could also be called the "and-configuration" of rules, into a "union configuration" that is theoretically equivalent to the and-configuration when the equivalence analysis is performed using crisp set theory[22] e.g., Combs and Andrews (1998), Weinschenk et al. (2003), Ross 2004). See Mendel and Liang (1999) for discussions about this method as well as replies to them by Andrews.
4. Combine input variables using sensor fusion [e.g., Jamshidi (1997, Sect. 8.3.2), Barai et al. (2015), Stover et al. (1996)]. Sensor fusion is the combining of sensory data or data derived from disparate sources such that the resulting information has less uncertainty than would be possible when these sources were used individually.
5. Use a hierarchical structure for the fuzzy system [e.g., Wang (1999)]. The following example illustrates this approach and points out that it seems to be very viable for controlling rule explosion.

Example 3.18 Wang (1999) provides the hierarchical architecture that is depicted in Fig. 3.19 in which the overall fuzzy system is a collection of $p-1$ TSK fuzzy sub-systems, each of which is described by a set of two-antecedent rules. The first fuzzy system has inputs x_1 and x_2, and its output is $y_1 = y_1(x_1, x_2)$; the second fuzzy system has inputs x_3 and y_1, and its output is y_2; the third fuzzy system has inputs x_4

[22]For example, for a two antecedents rule, it is true that $[(p \wedge q) \rightarrow r] \leftrightarrow [(p \rightarrow r) \vee (q \rightarrow r)]$ (see Exercise 2.35c).

and y_2, and its output is y_3; …; and, the $p-1$st fuzzy system has inputs x_p and y_{p-2}, and its output is y_{p-1} which is the overall output of the hierarchical system, i.e., $y_{p-1} = y$.

Wang (1999) proves universal approximation for this hierarchical architecture, and proves that this architecture is "optimal" in the sense that its total number of rules is minimal among all hierarchical fuzzy systems with p inputs. Wang (1997) also shows that the number of rules is a linear function of p, and provides this hierarchical architecture with the following very insightful observation:

> The problem of the standard fuzzy system is that the degree of freedom is unevenly distributed over the IF and THEN parts of the rules, with a very comprehensive IF part to cover the whole domain and a very simple THEN part. The hierarchical fuzzy system, on the other hand, tries to provide a balance between the IF and THEN parts, with an incomplete IF part but a more complex THEN part. Roughly speaking, the standard fuzzy system achieves universal approximation using "piecewise constant functions," while the hierarchical fuzzy system achieves universal approximation through "piecewise polynomial functions."

It would seem that this hierarchical fuzzy system would work best when its p inputs are first rank-ordered in importance (universal approximation does not address this important design issue).

Another hierarchical architecture is described in Sect. 4.4.7. It occurs because, even though there are p inputs, not all of them can be observed at any one time, something that is quite common when a fuzzy system is used as an aide for social judgment making.

3.9.5 Rule Interpretability

Because fuzzy systems are based on rules that use linguistic terms in their antecedents (and possibly in their consequents) they (or their associated basis functions) offer the unique feature of *interpretablility*. About interpretability, Gacto et al. (2011) state:

- Interpretability is the capacity to express the behavior of the real system in an understandable way. It is a subjective property that depends on the person who makes the assessment. This is related to several factors, mainly the model structure, the number of input variables, the number of fuzzy rules, the number of linguistic terms, the shape of the fuzzy sets, etc. There is still no standard measure to assess how good interpretability is.
- Accuracy and interpretability represent contradictory objectives.
- Due to its subjective nature and the large amount of factors involved, the choice of an appropriate interpretability measure is still an open problem. Most researchers would agree on interpretability involving aspects such as: The number of rules being as small as possible, rule premises that are easily understood in terms of structure and contain only a few input variables, linguistic terms that are intuitively comprehensible, etc.
- [there are] … two main kinds of approach to take into account the interpretability of linguistic fuzzy rule-based systems:

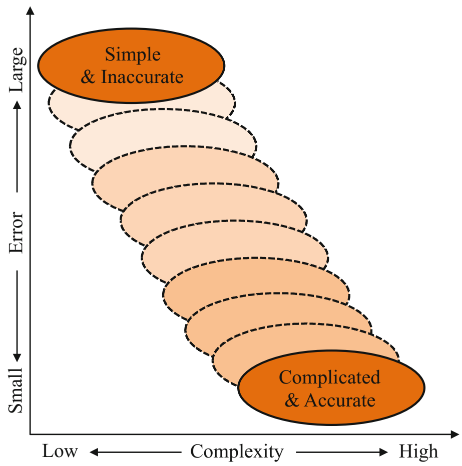


Fig. 3.20 Tradeoff between accuracy and complexity for a rule-based fuzzy system (Ishibuchi 2007, © 2007 IEEE)

– Complexity-based interpretability: these approaches are devoted to decreasing the complexity of the obtained model (usually measured as the number of rules, variables, labels per rule, etc.).
– Semantics-based interpretability: these approaches are devoted to preserving the semantics associated with the membership functions[23] … by imposing constraints on the MFs or … considering measures such as distinguishability, coverage, etc.

Cordon (2011) includes Fig. 3.20 [taken from Ishibuchi (2007)] that shows the tradeoff between accuracy and complexity. Obviously interpretable rules want to be at the low-end of complexity, for which there will be a high inaccuracy, and accurate rules lead to complicated fuzzy systems, in agreement with the Gacto et al. (2011) statement "accuracy and interpretability represent contradictory objectives."

Antonelli et al. (2015) state the following:

- Whereas in most applications "accuracy of prediction" is often the only metric used, in financial applications, interpretability and transparency are also important and sometimes a requirement. Within financial applications, the accuracy of the model is not the only crucial issue. There is a growing interest in high levels of model transparency, which is the ability to provide a clear and understandable explanation of the output results. … This need for transparency is reflected in legislation that forces financial institutions to disclose the reasoning behind their financial decisions and models. … Furthermore, transparency of a model is important because it allows users to understand data association by observing why a specific decision has been taken.
- Studies such as Casillas, et al. (2003), Setnes and Roubos (2000), Ishibuchi and Yamamoto (2003) and Ishibuchi et al. (2004) have shown that accuracy and interpretability are in a tradeoff and it is necessary to sacrifice one in order to increase the other. It is difficult to define to which extent accuracy or interpretability can be sacrificed in order to gain in the other. Usually different applications and specific situations have different requirements. Multi-objective genetic algorithms are able to provide an

[23]See, e.g., Herrera and Martinez (2000, 2001).

evolution through the two competitive objectives: accuracy and interpretability, as in Ducange and Marcelloni (2011), Fazzolari et al. (2013).

Other recent proponents for interpretability include Galende-Hernandez et al. (2012) and Garcia et al. (2015).

On the other hand, Hüllermeier (2015) is not a proponent for interpretability, and in his section entitled "The myth of interpretability" states:

> Interpretability is one of the core arguments often put forward by fuzzy scholars in favor of fuzzy models—usually in a very uncritical way. In fact, many authors seem to take it for granted that fuzzy models or, more specifically, fuzzy rule-based models, can easily be understood and interpreted by a human user or data analyst. Many of these authors apparently equate "fuzzy" with "linguistic" and "linguistic" with "interpretable," which, of course, is far too simple. A real "proof" of interpretability would require the presentation and careful inspection of a fuzzy model learned from data, which is almost never done. At best, a discussion of that kind is replaced by the computation of certain *interpretability measures*, which, however, are disputable and pretend to a level of objectivity that is arguably not warranted for this criterion.

About interpretability, this author tends to lean toward Hüllermeier when type-1 fuzzy sets are used, because (Hüllermeier also points out some of the following):

1. Words mean different things to different people, and, unless MFs are obtained by first collecting numerical word-data from a group of subjects who are representative of the end-users of the fuzzy system, there will be no direct connection between the words and their MFs. In fact, for interpretabililty, the words should come first and then their MFs, whereas for applications where interpretabililty is unimportant (e.g., fuzzy logic control, fuzzy system time-series forecasters, etc.), one can assign any names to the MFs, because it is only their formulas that are used to implement the input–output mapping of the fuzzy system. More is said about this in Sect. 4.4 and in Chap. 10.
2. People can only interpret/understand relatively simple rules because they have great difficulty in correlating more than two inputs at a time. More is said about this in Sect. 4.4.1 (item 5).

It is explained in Sect. 9.13.6 why fewer rules are usually needed when type-2 fuzzy sets are used, and this reduces complexity, so that (arguably) type-2 rules have a higher chance of being interpretable than do type-1 rules.

Exercises

3.1 Express the following Zadeh rule as a collection of complete IF-THEN Zadeh rules:
IF x_1 is moderate or x_2 is small and x_3 is very large, then, unless x_4 is big or x_5 is huge, THEN y is small

3.2 In Sect. 3.2, a DeMorgan's Law is used to convert an Unless rule to a complete IF-THEN rule. The usual proof of a DeMorgan's Law is for when

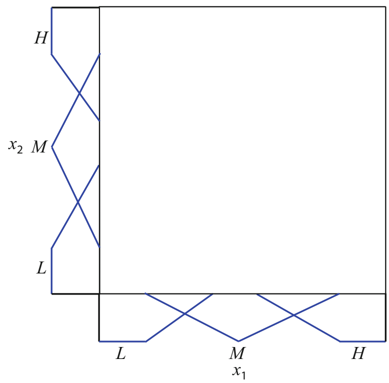


Fig. 3.21 Diagram for Exercise 3.3

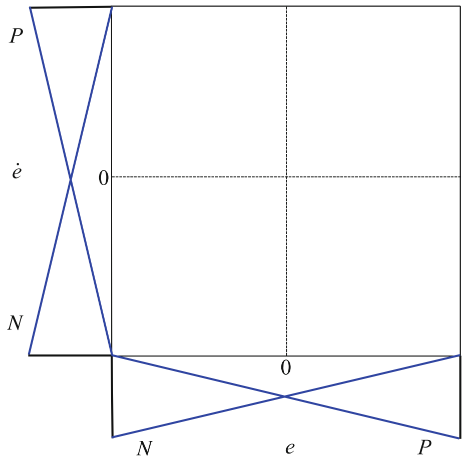


Fig. 3.22 Diagram for Exercise 3.4

all of the sets that are involved in it come from the same universe of discourse, but in the rule "IF x_1 is not F_1 or $\cdots$ or x_p is not F_p, THEN y is G", $x_1 \in X_1, \ldots, x_p \in X_p$, and $X_1, \ldots, X_p$ are often different. Prove DeMorgan's Law for this situation.

3.3 Repeat Example 3.3 for the MFs that are depicted in Fig. 3.21. Specifically (a) Obtain figures that are analogous to Figs. 3.3 and 3.4; (b) Determine how many type-1 first-order rule partitions there are; and, (c) Determine how many type-1 second-order rule partitions there are.

3.4 Repeat Example 3.3 for the MFs that are depicted in Fig. 3.22. Specifically: (a) Obtain figures that are analogous to Figs. 3.3 and 3.4; (b) Determine how many type-1 first-order rule partitions are there; (c) Determine how many type-1 second-order rule partitions are there; and, (d) How can the MFs be modified so that some second-order rule partitions occur?

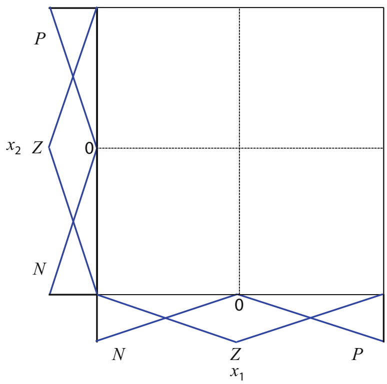


Fig. 3.23 Diagram for Exercise 3.5

3.5 Repeat Example 3.3 for the MFs that are depicted in Fig. 3.23. Specifically (a) Obtain figures that are analogous to Figs. 3.3 and 3.4; (b) Determine how many type-1 first-order rule partitions are there ; (c) Determine how many type-1 second-order rule partitions are there; and, (d) How can the MFs be modified so that some second-order rule partitions occur?

3.6 In Example 3.3, suppose the trapezoidal MFs are replaced by Gaussian MFs.

a) In theory, are there any type-1 first-order rule partitions?
b) In practice (i.e., truncate the Gaussians when their amplitude is less than epsilon), are there any type-1 first-order rule partitions, and, if so, what does a figure like Fig. 3.3 look like for them?

3.7 Provide figures that are comparable to Figs. 3.5, 3.6 and 3.9 but using triangle MFs. Do this for both minimum and product t-norms.

3.8 Provide figures that are comparable to Figs. 3.5, 3.8 and 3.9 but using triangle MFs. Do this for both minimum and product t-norms.

3.9 Create a graphical example of $\mu_B(y|\mathbf{x}')$ for which the mean-of-maxima defuzzifier provides a zero value even though $\mu_B(y|\mathbf{x}')$ is non-zero over most of its range.

3.10 Which features of a fired rule consequent set are used by height and center-of-sets defuzzifiers?

3.11 Repeat the Sect. 3.7 calculations for $\mathbf{x}' = col(0.5, -0.5)$.

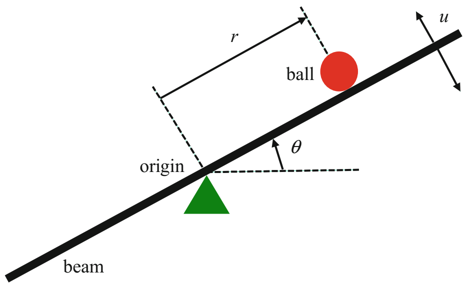
3.12 Show the type-1 first-order rule partitions of $X_1 \times X_2$ on a figure that is analogous to Fig. 3.3 for the MFs depicted in Fig. 3.14, and indicate how many rules are fired in each of the rule partitions.

3.13 Explain how $y_{TSK}^N(\mathbf{x}')$ in (3.54) is obtained.

3.14 Repeat Example 3.14 for the FBFs in (3.58).

3.15 [Adapted from Wang and Mendel (1992a, p. 811)] Fig. 3.24 depicts a ball on a beam. The beam is made to rotate in a vertical plane by applying a torque at the center of rotation and the ball is free to roll along the beam. The ball must

Fig. 3.24 Ball on beam (Wang and Mendel 1992a; © 1992 IEEE)



remain in contact with the beam. The control $u(t)$ is the acceleration of θ. The problem is to design a controller that drives the ball into the origin so that the ball remains at the origin. This design must be accomplished regardless of where the ball starts on the beam, and regardless of the position of the beam.

This system is nonlinear and is described by four state variables, $x_1 = r(t)$, $x_2 = dr(t)/dt$, $x_3 = \theta(t)$ and $x_4 = d\theta(t)/dt$. Four high-level linguistic supervisory control rules are given below:

$$\begin{cases} R_{Z1}^L : \text{IF } x_1 \text{ is } ''\text{positive}'' \text{ and } x_2 \text{ is } ''\text{near zero}'' \text{ and } x_3 \text{ is } ''\text{positive}'' \text{ and } x_4 \text{ is } ''\text{near zero}'', \\ \qquad\qquad \text{THEN } u \text{ is } ''\text{negative}'' \\ R_{Z2}^L : \text{IF } x_1 \text{ is } ''\text{positive}'' \text{ and } x_2 \text{ is } ''\text{near zero}'' \text{ and } x_3 \text{ is } ''\text{negative}'' \text{ and } x_4 \text{ is } ''\text{near zero}'', \\ \qquad\qquad \text{THEN } u \text{ is } ''\text{positive big}'' \\ R_{Z3}^L : \text{IF } x_1 \text{ is } ''\text{negative}'' \text{ and } x_2 \text{ is } ''\text{near zero}'' \text{ and } x_3 \text{ is } ''\text{positive}'' \text{ and } x_4 \text{ is } ''\text{near zero}'', \\ \qquad\qquad \text{THEN } u \text{ is } ''\text{negative big}'' \\ R_{Z4}^L : \text{IF } x_1 \text{ is } ''\text{negative}'' \text{ and } x_2 \text{ is } ''\text{near zero}'' \text{ and } x_3 \text{ is } ''\text{negative}'' \text{ and } x_4 \text{ is } ''\text{near zero}'', \\ \qquad\qquad \text{THEN } u \text{ is } ''\text{positive}'' \end{cases}$$

Here "positive" for x_1 is a fuzzy set $P1$, "negative" for x_1 is a fuzzy set $N1$, "near zero" for both x_2 and x_4 is a fuzzy set NZ, "positive" for x_3 is a fuzzy set $P3$, "negative" for x_3 is a fuzzy set $N3$, "positive" for u is a fuzzy set Pu, "negative" for u is a fuzzy set Nu, "positive big" for u is a fuzzy set PBu, and "negative big" for u is a fuzzy set NBu. The MFs for all of these fuzzy sets are given below:

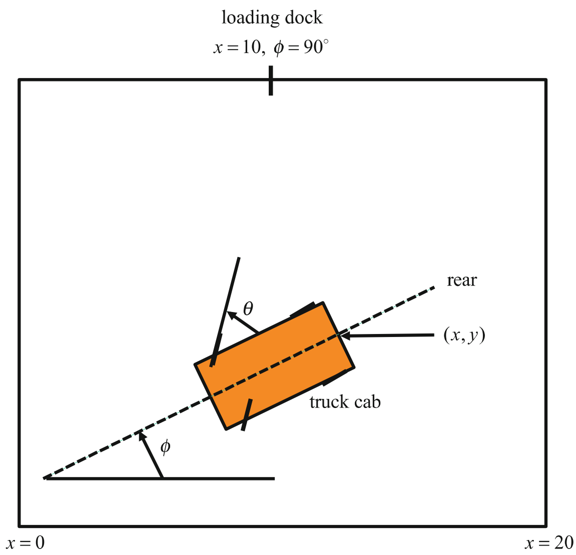


Fig. 3.25 Truck (cab) in relation to the loading dock (Mendel 1995; © 1995 IEEE)

$$\begin{cases} \mu_{P1}(x_1) = \exp[-\frac{1}{2}(\min(x_1 - 4, 0)/4)^2] \\ \mu_{N1}(x_1) = \exp[-\frac{1}{2}(\max(x_1 + 4, 0)/4)^2] \\ \mu_{NZ}(x) = \exp[-\frac{1}{2}x^2] \\ \mu_{P3}(x_3) = \exp[-\frac{1}{2}(\min(x_3 - \pi/4, 0)/(\pi/4))^2] \\ \mu_{N3}(x_3) = \exp[-\frac{1}{2}(\max(x_3 + \pi/4, 0)/(\pi/4))^2] \\ \mu_{Pu}(u) = \exp[-\frac{1}{2}(u - 0.1)^2] \\ \mu_{Nu}(u) = \exp[-\frac{1}{2}(u + 0.1)^2] \\ \mu_{PBu}(u) = \exp[-\frac{1}{2}(u - 0.4)^2] \\ \mu_{NBu}(u) = \exp[-\frac{1}{2}(u + 0.4)^2] \end{cases}$$

Write the FBF expansion for this Mamdani fuzzy system when the following are used: singleton fuzzification, product implication and COS defuzzification.

3.16 Wang and Mendel (1992b) Fig. 3.25 depicts a truck position to a loading dock. The truck's position is determined by the three state variables ϕ, x and y. Control to the truck is the angle θ . Only backing-up is considered. Enough clearance is assumed between the truck and the loading dock so that y does not have to be considered as an active state variable. The task is to design a control system whose inputs are $\phi \in [-90^\circ, 270^\circ]$ and $x \in [0, 20]$ and whose control is $\theta \in [-40^\circ, 40^\circ]$, such that the final state will be $(x_f, \phi_f) = (10, 90^\circ)$.

Assume that a collection of representative trajectories and control angles are obtained by collecting backing-up data from an experienced truck driver. From that data a set of 27 IF-THEN rules are obtained as is explained in Sect. 4.2.1.2. The rules are summarized in the relational matrix, given below. The entries in the matrix are the fuzzy sets for the control angle θ, which are a function of the two states, angular position, ϕ, and horizontal position, x. Blank entries have no consequent associated with them.

The MFs for ϕ are normal isosceles triangles with endpoints *a, b*, where: $(a_{S3} = -115°, b_{S3} = -15°)$, $(a_{S2} = -45°, b_{S2} = 45°)$, $(a_{S1} = 15°, b_{S1} = 90°)$, $(a_{CE} = 80°, b_{CE} = 100°)$, $(a_{B1} = 90°, b_{B1} = 165°)$, $(a_{B2} = 135°, b_{B2} = 225°)$, and $(a_{B3} = 195°, b_{B3} = 295°)$. The MFs for *S1*, *CE,* and *B1* of x are also normal isosceles triangles, where $(a_{S1} = 4, b_{S1} = 10)$, $(a_{CE} = 9, b_{CE} = 11)$, and $(a_{B1} = 10, b_{B1} = 16)$; *S2* and *B2* are left and right normal trapezoidal shoulder MFs, respectively, where (see Fig. 2.21): $S2 = (0, 0, 1.5, 7)$ and $B2 = (13, 18.5, 20, 20)$. The MFs for *S2*, *S1*, *CE*, *B1*, and *B2* of θ are also normal isosceles triangles, where $(a_{S2} = -33°, b_{S2} = -7°)$, $(a_{S1} = -14°, b_{S1} = 0°)$, $(a_{CE} = -4°, b_{CE} = 4°)$, $(a_{B1} = 0°, b_{B1} = 14°)$, and $(a_{B2} = 7°, b_{B2} = 33°)$; *S3* and *B3* are left and right normal trapezoidal (triangle) shoulder MFs, respectively, where $S3 = (-40°, -40°, -40°, -20°)$ and $B3 = (20°, 40°, 40°, 40°)$.

Relational matrix for Exercise 3.16

ϕ \ x	*S2*	*S1*	*CE*	*B1*	*B2*
S3	*S2*	*S3*			
S2	*S2*	*S3*	*S3*	*S3*	
S1	*B1*	*S1*	*S2*	*S3*	*S2*
CE	*B2*	*B2*	*CE*	*S2*	*S2*
B1	*B2*	*B3*	*B2*	*B1*	*S1*
B2		*B3*	*B3*	*B3*	*B2*
B3				*B3*	*B2*

Write the FBF expansion for this Mamdani fuzzy system when the following are used: singleton fuzzification, product implication, and height defuzzification.

3.17 Suppose one begins with the TSK FBF expansion in (3.56), and also has M_L expert Zadeh-rules. Obtain the counterparts to (3.67)–(3.69).

3.18 Arabshahi et al. (1996) provide nine rules, that are given in Table 3.5, for updating the learning factor β_θ in a steepest descent optimization algorithm. In that table, each entry represents the value of the fuzzy variable β_θ for given values of error (E) and change in error (CE), and, NS is *negative small*, ZE is *near zero*, and PS is *positive small*. An example of a rule is: IF the error is *medium* and the change in error is *high*, THEN the incremental update to β_θ is *near zero*. The three MFs for *E*, *CE,* and β_θ are shown in Fig. 3.26.

(a) Using singleton fuzzification, product t-norm, and COS defuzzification, provide a formula for $\beta_\theta(E, CE)$.

(b) Plot $\beta_\theta(E, CE)$ versus E and CE.

Table 3.5 Rules for Exercise 3.18

E\CE	*Low*	*Medium*	*High*
Low	PS	PS	ZE
Medium	PS	PS	ZE
High	ZE	ZE	NS

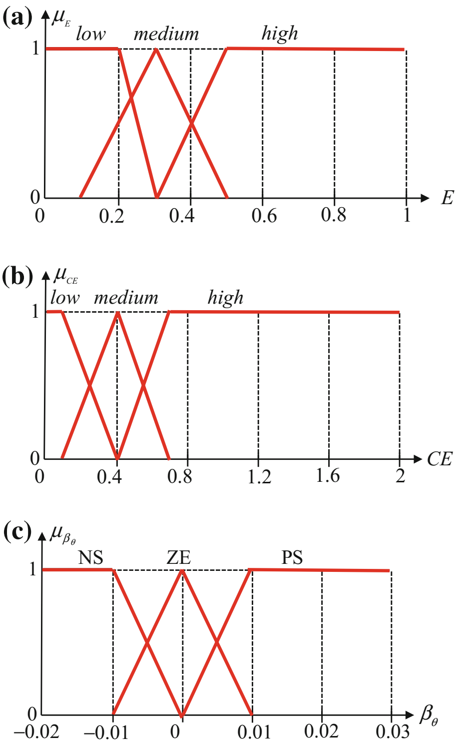


Fig. 3.26 MFs for Exercise 3.18. **a** Error, **b** change in error, and **c** learning factor

(c) Arabshahi et al. (1996) also provide a 25 rule fuzzy system for when β_θ is a function of *CE* and the change in *CE*, *CCE*, where both *CE* and *CCE* are described by five fuzzy sets. If you have access to this reference, repeat Parts a and b for this system. Compare $\beta_\theta(E, CE)$ and $\beta_\theta(CE, CCE)$, and draw some conclusions.

References

Alspach, D. L. and H. W. Sorenson. 1972. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transaction on Automatic Control*. 17: 439–448.

Antonelli, M. D. Bernardo, H. Hagras and F. Marcelloni. 2016. Multi-objective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification. Accepted for publication in *IEEE Transaction on Fuzzy Systems*.

Barai, R. K. T. Tjahjowidodo and B. K. Pappachan. 2015. Fuzzy inference system based intelligent sensor fusion for estimation of surface roughness in machine process, In *Proceeding of 9th IEEE international conference on sensing technology*, pp. 799–802, Aukland, New Zealand.

Blum, E.K., and L.K. Li. 1991. Approximation theory and feedforward networks. *Neural Networks* 4: 511–515.

Buckley, J.J. 1992. Universal fuzzy controllers. *Automatica* 28: 1245–1248.

Buckley, J.J. 1993. Sugeno-type-controllers are universal controllers. *Fuzzy Sets and Systems* 25: 299–303.

Casillas, J., O. Cordon, F. Herrera, and L. Magdalena (eds.). 2003. *Interpretability issues in fuzzy modeling*. Berlin Heidelberg: Springer-Verlag.

Castro, J. L. 1995. Fuzzy logic controllers are universal approximators, In *IEEE Transaction Systems on Man Cybernetics* 25(4): 629–635.

Combs, W.E., and J.E. Andrews. 1998. Combinatorial rule explosion eliminated by a fuzzy rule configuration. *IEEE Transaction on Fuzzy Systems* 6 (1): 1–11.

Cordon, O. 2011. A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal on Approximate Reason* 52: 894–913.

Cybenko, G. 1989. *Approximation by superpositions of a sigmoidal function*. Mathematics of Control: Signals Systems

Driankov, D. H. 1996. Hellendoorn and M. Reinfrank, *An Introduction to Fuzzy Control* (2nd ed.) Springer-Verlag.

Dubois, D., and H. Prade. 1985. A Review of fuzzy set aggregation connectives. *Info Science* 36: 85–121.

Ducange, P., and F. Marcelloni. 2011. Multi-objective evolutionary fuzzy systems", 9th international workshop on fuzzy logic applications, Italia, 29–31 August. *Lecture Notes on ComputerScience* 6857: 83–90.

Fazzolari, M., R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera. 2013. A review of the application of multi-objective evolutionary fuzzy systems: Current issues and further directions. *IEEE Transaction on Fuzzy Systems* 21 (1): 45–65.

Gacto, M.J., R. Alcala, and F. Herrera. 2011. Interpretability of linguistic fuzzy rule-based systems: an overview of interpretability measures. *Infomation Science* 181: 4340–4360.

Galende-Hernandez, M., G.I. Sainz-Palmero, and M.J. Fuente-Aparicio. 2012. Complexity reduction and interpretability improvement for fuzzy rule systems based on simple interpretability measures and indices by bi-objective evolutionary rule selection. *Soft Computing* 16: 451–470.

Garcia, D., J.C. Gamez, A. Gonzalez, and R. Perez. 2015. An interpretability improvement for fuzzy rule bases obtained by the iterative learning approach. *International Journal on Approximate Reason* 67: 37–58.

Haykin, S. 1996. *Adaptive filter theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall.

Herrera, F., and L. Martinez. 2000. A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transcation on Fuzzy Systems* 8 (6): 746–752.

Herrera, F. and L. Martinez. 2001. A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision-making, *IEEE Trans on Systems, Man, and Cybernetics, Part B (Cybernetics)* 31(2): 227–234.

Hornik, K. 1993. Some results on neural network approximation. *Neural Networks* 6: 1069–1072.

Hornik, K., M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2: 359–366.

Hüllermeier, E. 2015. Does machine learning need fuzzy logic? *Fuzzy Sets and Systems* 281: 292–299.

Ishibuchi, H. 2007. Multiobjective genetic fuzzy systems: Review and future research directions. In *Proceedings of 2007 IEEE international conference on fuzzy systems* (FUZZ-IEEE 2007), London, UK pp. 1–6.

Ishibuchi, H., T. Nakashima, and M. Nii. 2004. *Classification and modeling with linguistic information granules: Advanced approaches to linguistic data mining*. Berlin: Springer-Verlag.

Ishibuchi, H., and T. Yamamoto. 2003. Interpretability issues in fuzzy genetic-based machine learning for linguistic modeling. *Modeling With Words, Lecture Notes in Computer Sciences* 2873: 209–228.

Jamshidi, M. 1997. *Large Scale Systems: Modeling, Control and Fuzzy Logic,* Prentice-Hall, PTR, Upper Saddle River, NJ, (Section 8.3.2, Rule-base reduction).

Jang, J.-S. R. 1993. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transaction on Systems, Man and Cybernetics* 23: 665–684.

Juang, C.-F. 2002. A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and gnetic algorithm. *IEEE Transaction on Fuzzy System* 10(2): 155–170.

Juang, C.-F. and J.-S. Chen. 2006. Water bath temperature control by a recurrent fuzzy controller and its FPGA implementation. *IEEE Transaction on Industrial Electronics* 53(3): 941–949.

Juang, C.-F. and C.-T. Lin. 1999. A recurrent self-organizing neural fuzzy inference network. *IEEE Transcation on Neural Networks*, 10(4): 828–845.

Karnik, N. N. and J. M. Mendel. 1998. *An Introduction to Type-2 Fuzzy Logic Systems*, USC-SIPI Report #418, Univ. of Southern Calif., Los Angeles, CA, June 1998. This can be accessed at: http://sipi.usc.edu/research; then choose "sipi technical reports/418".

Keller, J.M., R.R. Yager, and H. Tahani. 1992. Neural network implementation of fuzzy logic. *Fuzzy Sets and Systems* 45: 1–12.

Kim, Y. M. and J. M. Mendel. 1995. Fuzzy basis functions: Comparisons with other basis functions. *IEEE Transcation on Fuzzy Systems* 3: 158–168.

Kiska, J. B. M. E. Kochanska and D. S. Sliwinska. 1985a. The influence of some parameters on the accuracy of fuzzy model. In *Industrial applications of fuzzy control* (M. Sugeno, Ed.), North Holland, Amsterdam, pp. 187–230.

Kiska, J.B., M.E. Kochanska, and D.S. Sliwinska. 1985b. The influence of some parameters on the accuracy of a fuzzy model—part I. *Fuzzy Sets and Systems* 15: 111–128.

Kiska, J.B., M.E. Kochanska, and D.S. Sliwinska. 1985c. The influence of some parameters on the accuracy of a fuzzy model—part II. *Fuzzy Sets and Systems* 15: 223–240.

Kóczy, L. T, and K. Hirota. 1997. Size reduction by interpolation in fuzzy rule bases, *IEEE Transcation on Systems and Man Cybernetics—Part B: Cybernetics* 27: 14–25.

Kosko, B. 1992. Fuzzy Systems as Universal Approximators. *IEEE International Conference on Fuzzy Systems,* pp. 1153–1162.

Kosko, B. 1994. Fuzzy systems as universal approximators. *IEEE Transcation on Computers* 43 (11): 1329–1333.

Kosko, B. 1997. *Fuzzy engineering*. Upper Saddle River, NJ: Prentice Hall.

Kovács, S. 2009. Fuzzy Rule Interpolation. In *Encyclopedia of Artificial Intelligence,* ed. Juan Ramón Rabuñal Dopico, Julian Dorado and Alejandro Pazos, pp. 728–733, 2009, accessed January 15, 2016. doi:10.4018/978-1-59904-849-9.ch108.

Kreinovich, V., G.C. Mouzouris, and H.T. Nguyen. 1998. Fuzzy rule based modeling as a universal approximation tool. In *Fuzzy systems, modeling and control*, ed. T. Nguyen, and M. Sugeno, 135–195. Boston: H Kluwer Ac. Publ.

Kreinovich, V., H.T. Nguyen, and Y. Yam. 1999. Fuzzy systems are universal approximators for a smooth function and its derivatives. *International Journal of Intelligent Systems* 15 (6): 565–574.

Lee, C.-C. 1990. Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II. *IEEE Transcation on System and Man Cybernetics* 20: 419–435.

Lee, C.-H. and C.-C. Teng. 2000. Identification and control of dynamic systems using recurrent fuzzy neural networks. *IEEE Transaction on Fuzzy Systems* 8(4): 349–366.

Lin, C.-J and C.-C. Chin. 2004. Prediction and identification using wavelet-based recurrent fuzzy neural networks. *IEEE Transcation Systems and Man Cybernetics Part B: Cybernetics*34(5): 2144–2154.

Lin, C.-T. and C. S. G. Lee. 1991. Neural-network-based fuzzy logic control and decision system. *IEEE Transcation on Computers* 40(12): 1320–1336.

Lin, C.-T., and C.S.G. Lee. 1996. *Neural fuzzy systems: A neuro-fuzzy synergism to intelligent systems*. Upper Saddle River, NJ: Prentice-Hall PTR.

Mendel, J. M. 1995. Fuzzy logic systems for engineering: A tutorial. *IEEE Proceedings* 83: 345–377.

Mendel, J.M. 2001. *Introduction to rule-based fuzzy logic systems*. Upper Saddle River, NJ: Prentice-Hall.

Mendel, J. M. and Q. Liang. 1999. Comments on combinatorial rule explosion eliminated by a fuzzy rule configuration. by W. E. Combs and J. E. Andrews, *IEEE Transaction on Fuzzy Systems*, vol. 7, pp. 369–373, June 1999.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: aiding people in making subjective judgments*. Hoboken, NJ: Wiley and IEEE Press.

Mizumoto, M. 1987. Comparison of various fuzzy reasoning methods. *Proceeding of 2nd IFSA Congress* Tokyo, Japan, pp. 2–7.

Moody, J., and C.J. Darken. 1989. Fast learning in networks of locally-tuned processing units. *Neural Computing* 1: 281–294.

Mouzouris, G.C., and J.M. Mendel. 1996. In *Proceedings on fifth IEEE international conference on fuzzy systems designing fuzzy logic systems for uncertain environments using a singular-value–QR decomposition method*. New Orleans, LA:

Mouzouris, G. C. and J. M. Mendel. 1997. Non-singleton fuzzy logic systems: Theory and applications. *IEEE Transaction on Fuzzy Systems* 5: 56–71.

Mouzouris, G.C., and J.M. Mendel. 1997b. A singular-value–QR decomposition based method for training fuzzy logic systems in uncertain environments. *Journal on Intelligent and Fuzzy Systems* 5: 367–374.

Poggio, T. and F. Girosi. 1990. Networks for approximation and learning. *Proceeding of IEEE*, vol. 78, pp. 1481–1497.

Raha, S., N. R. Pal and K. S. Ray. 2002. Similarity based approximate reasoning: methodology and application. *IEEE Transaction on Systems, Man and Cybernetics A* 32(2): 541–547.

Raha, S., A. Hossain, and S. Ghosh. 2008. Similarity based approximate reasoning: Fuzzy control. *Journal on Applied Logic* 6: 47–71.

Ross, T.J. 2004. *Fuzzy logic with engineering applications*, 2nd ed. Boston: Wiley.

Ruspini, E. 1969. A new approach to clustering. *Info Control* 15: 22–32.

Setnes, M., and H. Roubos. 2000. GA-fuzzy modeling and classification: Complexity and performance. *IEEE Transaction on Fuzzy System* 8 (5): 509–522.

Specht, D.F. 1991. A general regression neural network. *IEEE Transcation on Neural Networks* 2: 568–576.

Stachowicz, M. S. and M. E. Kochanska. 1987. Fuzzy Modeling of the Process. In *Proceedings on 2nd IFSA congress*, Tokyo, Japan, pp. 86–89.

Stover, J. A., D. L. Hall and R. E. Gibson. 1996. A fuzzy-logic architecture for autonomous multi-sensor data fusion. *IEEE Transaction on Industrial Electronics* 43: 403–410.

Sugeno, M., and G.T. Kang. 1988. Structure identification of fuzzy model. *Fuzzy Sets and Systems* 28: 15–33.

Sugeno, M., and T. Yasukawa. 1993. A fuzzy-logic-based approach to qualitative modeling. *IEEE Transaction on Fuzzy System* 1: 7–31.

Takagi, T. and M. Sugeno. 1985. Fuzzy identification of systems and its application to modeling and control. *IEEE Transcation on Systems Man Cybernetics*, 15: 116–132.

Tanaka, K., M. Sano, and H. Watanabe. 1995. Modeling and control of carbon monoxide concentration using a neuro-fuzzy technique. *IEEE Transaction on Fuzzy System* 3: 271–279.

Tanaka, K., and M. Sugeno. 1998. Introduction to fuzzy modeling. In *Fuzzy systems modeling and control*, ed. T. Nguyen, and M. Sugeno, 63–89. Boston, MA: H Kluwer Academic Publ.

Tanaka, K. and H. O. Wang. 2001. *Fuzzy control systems design and analysis: A linear matrix inequality approach*, Wiley Interscience.

Tao, K. M. 1993. A closer look at the radial basis function (RBF) networks. In *Proceeding of 27th Asilomar conference on signals, systems and computers*, Pacific Grove, CA.

Theocharis, J.B. 2006. A high-order recurrent neuro-fuzzy system with internal dynamics: Application to adaptive noise cancellation. *Fuzzy Sets and Systems* 157 (4): 471–500.

Tsang, F.C.C., J.W.T. Lee, and D.S. Yeung. 1995. *Similarity based fuzzy reasoning methods for fuzzy production rules in Procedings*, 157–160. World Congress: IFSA.

Türksen, I. B. and Z. Zhong. 1988. An approximate analogical reasoning approach based on similarity measures. *IEEE Transcation on Systems, Man and Cybernetics* 18(6): 1049–1056.

Vadiee, N. and M. Jamshidi. 1993.A tutorial on fuzzy rule-based expert systems (FRBES) models. 1: Mathematical foundations. *Journal on Intelligent and Fuzzy Systems* 1: 171–188.

Wagner, C., A. Pourabdollah, J. McCulloch, R. John and J. Garibaldi. 2016. A similarity-based inference engine for non-singleton fuzzy logic systems. In *Proceeding of IEEE international conference on fuzzy systems*, Vancouver, Canada, pp. 316–323.

Wang, K. Y., D. Tikk, T. D. Gedeon and L. T. Koczy. 2005. Fuzzy rule interpolation for multidimensional spaces with applications: A case study. *IEEE Transaction on Fuzzy System* 13(6): 809–819.

Wang, L.-X. 1992a. *Analysis and design of fuzzy systems*, Ph.D. Dissertation, University of Southern California, Los Angeles, CA.

Wang, L.-X. 1992b. *Fuzzy systems are universal approximators*. In *Proceedings on IEEE international conference on fuzzy systems* San Diego, CA.

Wang, L.-X. 1994. *Adaptive fuzzy systems and control: Design and stability analysis*. Englewood Cliffs, NJ: PTR Prentice-Hall.

Wang, L.-X. 1997. *A course in fuzzy systems and control*. Upper Saddle River, NJ: Prentice-Hall.

Wang L.-X. 1999. Analysis and design of hierarchical fuzzy systems. *IEEE Transaction on Fuzzy System* 7: 617–624.

Wang, L.-X. and J. M. Mendel. 1992a. Fuzzy basis functions, universal approximation, and orthogonal least squares learning. *IEEE Transcation on Neural Networks* 3: 807–813.

Wang, L.-X. and J. M. Mendel. 1992b Generating fuzzy rules by learning from examples. *IEEE Transcation on Systems and Man Cybernetics* 22: 1414–1427.

Wang, Y.-C., C.-J. Chien, and C.-C. Teng. 2004. Direct adaptive iterative learning control of nonlinear systems using an output-recurrent fuzzy neural network. *IEEE Transcation on Systems and Man Cybernetics-B: Cybernetics*, 34(3): 1348–1359.

Weinschenk, J.J., W.E. Combs, and R.J. Marks II. 2003. In *Proceedings on IEEE international conference on fuzzy systems avoidance of rule explosion by mapping fuzzy systems to a union rule configuration* , 43–48. St. Louis MO.

Wu, D. and J. M. Mendel. 2009. Perceptual reasoning for perceptual computing: A similarity-based approach. *IEEE Transactions Fuzzy Systems* 17: 1397–1411.

Wu, D., and J.M. Mendel. 2011. On the continuity of type-1 and interval type-2 fuzzy logic systems. *IEEE Transaction on Fuzzy System* 19 (1): 179–192.

Yam, Y. P. Baranyi and C.-T. Yang. 1999. Reduction of fuzzy rule base via singular value decomposition. *IEEE Transaction on Fuzzy System* 7: 120–132.

Yen, J., and R. Langari. 1999. *Fuzzy logic: Intelligence, control, and information*. Upper Saddle River, NJ: Prentice-Hall.

Yen, J., and L. Wang. 1996. *An SVD-based fuzzy model reduction strategy*. In *Proceedings of the fifth international conference on fuzzy systems*, 835–841. New Orleans, LA.

Yen, J. and L. Wang. 1999. Simplifying fuzzy rule-based models using orthogonal transformations. *IEEE Transaction Systems Man and Cybernetics*.

Zadeh, L. A. 1973. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transaction Systems Man Cybernetics* vol. SMC-3, pp. 28–44.

Zhang, J. and A. J. Morris. 1999. Recurrent neuro-fuzzy networks for nonlinear process modeling. *IEEE Transaction Neural Networks* 10(2): 313–326.

Chapter 4
Type-1 Fuzzy Systems: Design Methods and Applications

4.1 Designing Type-1 Fuzzy Systems

From the detailed discussions given about the four elements that comprise the Fig. 3.1 fuzzy system, it should be clear that there are many possibilities to choose from. Table 4.1 enumerates them. Not included in that table is which input variables should be used, because this is application dependent. For some applications the choice of which variables to use is easy because of historical precedence (e.g., for fuzzy logic control one would choose the system's states). For others, the choice may be very challenging because there may be no historical precedence (e.g., for time series forecasting one would choose delayed versions of the main variable, however, how many delays to choose is usually unknown and may need to be varied). In the latter situation, it is best to seek the advice of experts, or, if none are available, to create more than one fuzzy system in order to see which choices lead to a best design.

Although Table 4.1 should be self-explanatory, two examples are presented next that illustrate its use.

Example 4.1 Table 4.2 is for a Mamdani fuzzy system which uses singleton fuzzification, product t-norm, COS defuzzification (so fired-rule output sets are not combined), four input variables, three terms for each variable all of which are named *low*, *moderate* and *high*, four output terms named *none to very little*, *some*, *a moderate amount* and *a lot*, 81 rules (3^4), and triangle MFs whose parameters will be optimized.

Example 4.2 Table 4.3 is for a TSK fuzzy system which uses non-singleton fuzzification, minimum t-norm, normalized defuzzification, three input variables, three terms for each variable all of which are named *light*, *moderate* and *heavy*, a consequent function for each rule that is a linear combination of the three input variables, 27 rules (3^3), and Gaussian MFs whose parameters are prespecified.

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_4

Table 4.1 Choices that need to be made to specify or design a type-1 fuzzy system

Choices for	Kind of type-1 fuzzy system							
• Rules	Mamdani				TSK			
• Fuzzifier	Singleton		Non-singleton		Singleton		Non-singleton	
• t-norm	Minimum		Product		Minimum		Product	
• Combining fired rule output sets	Union	Additive	None		NA			
• Defuzzifier	Centroid	Height	COS		Unnormalized		Normalized	
• Number of input variables	p				p			
• Number and names of terms per input	$\{X_{1j}\}_{j=1}^{Q_1}$	$\{X_{2j}\}_{j=1}^{Q_2}$	$\cdots$	$\{X_{pj}\}_{j=1}^{Q_p}$	$\{X_{1j}\}_{j=1}^{Q_1}$	$\{X_{2j}\}_{j=1}^{Q_2}$	$\cdots$	$\{X_{pj}\}_{j=1}^{Q_p}$
• Number and names of terms for, or structure of, output	$\{Y_j\}_{j=1}^{Q_y}$				$\{g^l(\mathbf{x})\}_{l=1}^{M}$			
• Number of rules	M				M			
• Kind of MFs	Gaussian	Triangle	Trapezoid	Other	Gaussian	Triangle	Trapezoid	Other
• MF parameters	Pre-specified		Optimized		Pre-specified		Optimized	

Table 4.2 Example of choices (shaded) made to specify or design a Mamdani type-1 fuzzy system

Choices for	Kind of type-1 fuzzy system							
• Rules	Mamdani				TSK			
• Fuzzifier	Singleton		Non-singleton		Singleton		Non-singleton	
• t-norm	Minimum		Product		Minimum		Product	
• Combining fired rule output sets	Union	Additive	None		NA			
• Defuzzifier	Centroid	Height	COS		Unnormalized		Normalized	
• Number of input variables	4				p			
• Number and names of terms per input	*low*	*moderate*	*high*		$\{X_{1j}\}_{j=1}^{Q_1}$	$\{X_{2j}\}_{j=1}^{Q_2}$	$\cdots$	$\{X_{pj}\}_{j=1}^{Q_p}$
• Number and names of terms for, or structure of, output	*none to very little*	*some*	*a moderate amount*	*a lot*	$\{g^l(\mathbf{x})\}_{l=1}^{M}$			
• Number of rules	81				M			
• Kind of MFs	Gaussian	Triangle	Trapezoid	Other	Gaussian	Triangle	Trapezoid	Other
• MF parameters	Pre-specified		Optimized		Pre-specified		Optimized	

Table 4.3 Example of choices (shaded) made to specify or design a TSK type-1 fuzzy system

Choices for	Kind of type-1 fuzzy system	
• Rules	Mamdani	TSK
• Fuzzifier	Singleton, Non-singleton	Singleton, Non-singleton
• t-norm	Minimum, Product	Minimum, Product
• Combining fired rule output sets	Union, Additive, None	NA
• Defuzzifier	Centroid, Height, COS	Unnormalized, Normalized
• Number of input variables	p	3
• Number and names of terms per input	$\{X_{1j}\}_{j=1}^{Q_1}$, $\{X_{2j}\}_{j=1}^{Q_2}$, …, $\{X_{pj}\}_{j=1}^{Q_p}$	*light*, *moderate*, *heavy*
• Number and names of terms for, or structure of, output	$\{Y_j\}_{j=1}^{Q_y}$	$\{c_0^l + c_1^l x_1 + c_2^l x_2 + c_3^l x_3\}_{l=1}^{M}$
• Number of rules	M	27
• Kind of MFs	Gaussian, Triangle, Trapezoid, Other	Gaussian, Triangle, Trapezoid, Other
• MF parameters	Pre-specified, Optimized	Pre-specified, Optimized

Ultimately, after one has made all of the choices indicated in Table 4.1, the parameters of the MFs must be fixed. Prior to 1992, all fuzzy systems reported in the open literature fixed these parameters somewhat arbitrarily, e.g., the locations and spreads of the MFs were chosen by the designer independent of the numerical training data. Then, at the first IEEE Conference on Fuzzy Systems, held in San Diego in 1992, three different groups of researchers, Horikowa et al. (1992), Jang (1992) and Wang and Mendel (1992b), presented the same idea: *tune the parameters of a fuzzy system using the numerical training data*. Since that time, quite a few adaptive training procedures have been published. Because tuning of free parameters had been done in feedforward neural networks long before it was done in a fuzzy system, a tuned fuzzy system has also come to be known as a *neural–fuzzy system*, even though, as explained in Sect. 3.9.1 there is nothing neural about a fuzzy system.

Designing a fuzzy system (Mendel and Mouzouris 1997) can be viewed as approximating a function or fitting a complex surface in a multidimensional high-dimensional space. Given a set of input–output pairs, tuning is essentially equivalent to determining a system that provides an optimal fit to the input–output pairs, with respect to a cost function. In addition, the system produced by the tuning algorithm should be able to generalize to certain regions of the multidimensional space where no training data are given, i.e., it must be able to interpolate the given input–output data. Within the framework of approximation and interpolation theory, it is common in many approximation/interpolation methods to generate the desired surface using a linear combination of basis functions (typically nonlinear transformations of the input). It has already been seen, in Sect. 3.8, that a fuzzy system can be expressed as such a linear combination of fuzzy basis functions (FBFs).

There exists a multitude of design methods that can be used to construct fuzzy systems that have different properties and characteristics. Some of these design methods are data-intensive, some are aimed at computational simplicity, some are recursive (thus giving the fuzzy system an adaptive nature), some are offline, and some are application-specific. The textbook (Lin and Lee 1996) has a very good summary of many of the methods that were published prior to 1996.[1]

The goal of this section is *not* to describe the different design approaches in complete detail. Instead, some of them are described briefly in connection with the following two problems:

1. Given N input–output numerical data pairs, $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots,$ $(\mathbf{x}^{(N)} : y^{(N)})$, where $\mathbf{x}$ is the vector input and y is the scalar output of a singleton fuzzy system, completely specify a singleton Mamdani or TSK fuzzy system using this data.

[1]The first edition of this book (Mendel 2001, p. 158, footnote 3) provided more than 35 references, but that was more than 16 years ago. Since that time many more design methods have been published. With the advent of the Internet, it is now very easy for a reader to find such references; hence, they are not included in this edition of the book.

2. Given N input–output numerical data pairs, $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots, (\mathbf{x}^{(N)} : y^{(N)})$, where $\mathbf{x}$ is the vector *noisy* input and y is the scalar (possibly) *noisy* output of a non-singleton fuzzy system, completely specify a non-singleton Mamdani or TSK fuzzy system using this data.

For illustrative purposes, *all designs in this chapter assume Gaussian MFs*. The Mamdani fuzzy system is described by (3.55) and (3.57), and the TSK fuzzy system is described by (3.56) and (3.59). These equations are repeated next for the convenience of the readers.

Mamdani fuzzy system:

$$y(\mathbf{x}) = \sum_{l=1}^{M} \lambda^l \phi_l(\mathbf{x}) \tag{4.1}$$

$$\phi_l(\mathbf{x}) = \begin{cases} \dfrac{\prod_{i=1}^{p} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i^l}(x_i)} & \text{singleton fuzzification} \\ \dfrac{\prod_{k=1}^{p} \mu_{Q_k^l}\left(x_{k,\max}^l \middle| x_i\right)}{\sum_{l=1}^{M} \prod_{k=1}^{p} \mu_{Q_k^l}\left(x_{k,\max}^l \middle| x_i\right)} & \text{non-singleton fuzzification} \end{cases} \tag{4.2}$$

TSK fuzzy system:

$$y_{\mathrm{TSK}}(\mathbf{x}) = \sum_{l=1}^{M} \sum_{j=0}^{p} c_j^l \phi_j^l(\mathbf{x}) \tag{4.3}$$

$$\phi_j^l(\mathbf{x}) = \begin{cases} \dfrac{\prod_{i=1}^{p} x_j \times \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i^l}(x_i)} & \text{singleton fuzzification} \\ \dfrac{\prod_{i=1}^{p} x_j \times \mu_{Q_i^l}\left(x_{i,\max}^l \middle| x_i\right)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{Q_i^l}\left(x_{i,\max}^l \middle| x_i\right)} & \text{non-singleton fuzzification} \end{cases} \tag{4.4}$$

In (4.4), $x_0 \equiv 1$, and in (4.2) and (4.4) $(i = 1, \ldots, p;\ l = 1, \ldots, M)$:

$$\mu_{F_i^l}(x_i) = \exp\left\{-\tfrac{1}{2}\left[(x_i - m_{F_i^l})/\sigma_{F_i^l}\right]^2\right\} \tag{4.5}$$

$$\mu_{Q_i^l}\left(x_{i,\max}^l \middle| x_i'\right) = \exp\left\{-(x_i' - m_{F_i^l})^2 / 2\left(\sigma_{X_i}^2 + \sigma_{F_i^l}^2\right)\right\} \equiv \mu_{Q_i^l}(x_i') \tag{4.6}$$

Each design method establishes how to specify the fuzzy system's MF and consequent parameters using the data pairs $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots, (\mathbf{x}^{(N)} : y^{(N)})$. Before describing some design methods, it is important to relate the number of rules, M, and the number of data samples, N, to one another.

If no tuning of the fuzzy system's parameters is done, then there can be as many as $M = N$ rules. If tuning is used, and one abides by the commonly used design principle that there must be fewer tuned design parameters than data pairs, then $M < N$. The exact inequality relationship between M and N depends on many choices, as is explained next.

Example 4.3 Table 4.4 summarizes the parameters that have to be optimized for each kind of a fuzzy system, and Table 4.5 summarizes the maximum number of design parameters. For example, in a Mamdani singleton fuzzy system, there are 2 parameters per antecedent; hence, for p antecedents and M rules, there will be a total of $2pM$ antecedent parameters. Additionally, there is one consequent parameter per rule; hence, for M rules there will be a total of M consequent parameters. Consequently, the maximum number of design parameters is $2pM + M$, "maximum" because in some design methods all of these parameters are not tuned, i.e., some are fixed ahead of time.

In other design approaches the number of parameters may be reduced. For example, in Jang (1993) and Jang and Sun (1995), each rule has p antecedents and each antecedent has a fixed number (s) of fuzzy sets that are common to all M rules, and it is assumed that each of the M rule's common antecedent MFs are characterized by the *same* two parameters; hence, there will only be $2ps$ antecedent parameters—and $2ps \ll 2pM$.

From the discussion given just before this example about the number of rules versus the number of data pairs, one can conclude that, in this example, if tuning is used, then the following design constraints must hold:

Table 4.4 Parameters that need to be optimized in a fuzzy system $(l = 1, \ldots, M)$

Parameters	Kind of type-1 fuzzy system			
	Mamdani	Mamdani	TSK	TSK
	Singleton	Non-singleton	Singleton	Non-singleton
Antecedent	$\{m_{F_i^l}, \sigma_{F_i^l}\}_{i=1}^p$	$\{m_{F_i^l}, \sigma_{F_i^l}\}_{i=1}^p$	$\{m_{F_i^l}, \sigma_{F_i^l}\}_{i=1}^p$	$\{m_{F_i^l}, \sigma_{F_i^l}\}_{i=1}^p$
Fuzzifier	–	$\{\sigma_{X_i}\}_{i=1}^p$	–	$\{\sigma_{X_i}\}_{i=1}^p$
Consequent	λ^l	λ^l	$\{c_j^l\}_{j=0}^p$	$\{c_j^l\}_{j=0}^p$

Table 4.5 Maximum number of parameters that need to be optimized in a fuzzy system

Parameters	Kind of type-1 fuzzy system			
	Mamdani	Mamdani	TSK	TSK
	Singleton	Non-singleton	Singleton	Non-singleton
Antecedent	$2pM$	$2pM$	$2pM$	$2pM$
Fuzzifier	0	p	0	p
Consequent	M	M	$(p+1)M$	$(p+1)M$
Maximum number	$(2p+1)M$	$(2p+1)M+p$	$(3p+1)M$	$(3p+1)M+p$

$$\begin{cases} (2p+1)M < N & \text{Mamdani singleton fuzzy system} \\ (2p+1)M + p < N & \text{Mamdani non-singleton fuzzy system} \\ (3p+1)M < N & \text{TSK singleton fuzzy system} \\ (3p+1)M + p < N & \text{TSK non-singleton fuzzy system} \end{cases} \tag{4.7}$$

Usually, the number of antecedents, p, is fixed ahead of time; so, the inequalities in (4.7) can be used to choose M or N. For example, if one is given a fixed number of $N = N'$ data samples, then

$$\begin{cases} M < N'/(2p+1) & \text{Mamdani singleton fuzzy system} \\ M < (N'-p)/(2p+1) & \text{Mamdani non-singleton fuzzy system} \\ M < N'/(3p+1) & \text{TSK singleton fuzzy system} \\ M < (N'-p)/(3p+1) & \text{TSK non-singleton fuzzy system} \end{cases} \tag{4.8}$$

(4.8) constrains the number of rules that can be used. Because M must be an integer, one chooses $M = M'$ as an integer that is smaller than the right-hand sides of the inequalities that are given in (4.8).

At a very high level, one can think of a variety of designs associated with the preceding formulations. For example, *for Mamdani and TSK <u>singleton</u> fuzzy systems*:

1. Fix the shapes and parameters of all antecedent and consequent MFs (or consequent functions) ahead of time. The data establish the rules and no tuning is used.
2. Fix the shapes and parameters of the antecedent MFs ahead of time. Use the data pairs to tune only the consequent parameters [the λ^l in (4.1) or the $\{c_0^l, c_1^l \ldots, c_p^l\}_{l=1}^M$ in (4.3)]. In this case, the fuzzy systems are linear in the consequent parameters, so the optimization problem is easy to solve.
3. Fix the shapes of all antecedent and consequent MFs ahead of time. Use the data pairs to tune the antecedent and consequent parameters. Because the fuzzy systems are nonlinear in the antecedent parameters [see (4.2) and (4.4)], the optimization problem is more difficult to solve than in the second design.

Similarly, for *Mamdani and TSK <u>non-singleton</u> fuzzy systems*:

1. Fix the shapes and parameters of all antecedent, consequent, and input measurement MFs (or consequent functions) ahead of time. The data establish the rules and the standard deviation of the measurements, and no tuning is used.
2. Fix the shapes and parameters of the antecedent and input measurement MFs ahead of time. Use the data pairs to tune only the consequent parameters. Because the fuzzy systems are linear in the consequent parameters, the optimization problem is easy to solve.

3. Fix the shapes and parameters of all antecedent and consequent MFs (or consequent functions) ahead of time. Fix the shape but not the parameter(s) of the input measurement MFs ahead of time. Use the data pairs to tune the parameters of the input measurement MFs. Because the fuzzy systems are nonlinear in these parameters, the optimization problem is more difficult to solve than in the second design.
4. Fix the shapes of all antecedent, consequent, and input measurement MFs (or consequent functions) ahead of time. Use the data pairs to tune the antecedent, consequent, and input measurement parameters. Because the fuzzy systems are nonlinear in both the antecedent and input measurement MF parameters, the optimization problem is even more difficult to solve than in the third design.

Naturally, other even more complicated design problems can be formulated that try to answer questions such as: (a) What shape MFs should be used?; (b) How many and which antecedents should be used (e.g., can p be reduced?)?; and, (c) How many and which rules should be used (e.g., can M be reduced?)? Evolutionary optimization algorithms can be used to address the first of these questions, systematic trial and error or evolutionary optimization algorithms are viable approaches to answering the second question, and singular-value decomposition techniques and evolutionary optimization algorithms can be used to answer the third question. Because answers to the second and third questions can reduce the complexity of the fuzzy system, these questions should not be ignored.

Before briefly touching on some design methods that can be used for the different designs just formulated, note that there can be two very different *approaches to the tuning of a non-singleton fuzzy system*:

1. **Partially dependent approach**: In this approach, one first designs the best possible singleton fuzzy system, by tuning all of its parameters, and then updates the design by: (a) keeping all of the parameters that are shared by the singleton and non-singleton fuzzy systems fixed at the values obtained from the best possible singleton fuzzy system, and (b) tuning only the new parameter(s) of the non-singleton fuzzy system. In the present case, only the standard deviations, σ_{X_i}, would be tuned.
2. **Totally independent approach**: In this approach, all parameters of the non-singleton fuzzy system are tuned. If, perchance, a singleton fuzzy system has already been designed, then its parameters can be used as the *initial parameters* for the tuning algorithms of the parameters that are shared by the singleton and non-singleton fuzzy systems.

One would expect the best performance to be obtained by the totally independent approach; however, sometimes it is very useful to use the partially dependent approach to observe the incremental improvement that can be obtained from the previous design to the new design.

4.2 Some Design Methods

This section briefly describes some design methods that can be used for the fuzzy system designs that were just formulated in Sect. 4.1. Some involve no tuning of MF parameters and others involve lots of tuning—optimization—of those parameters.

4.2.1 *One-Pass Methods*

A one-pass method is a method that uses the data one time and goes directly from it to Zadeh rules. In the rest of this section, two one-pass methods are described.

4.2.1.1 Data Assignment Method

In this method, the data pairs establish the centers of the fuzzy sets that appear in the antecedents and consequents of the rules. Here, for example, are N rules that can be extracted from the N data pairs, $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots, (\mathbf{x}^{(N)} : y^{(N)})$ (Mendel 1995a):

1. R^1: IF x_1 is F_1^1 and x_2 is F_2^1 and … and x_p is F_p^1, THEN y is G^1—In this rule, which is obtained from $(\mathbf{x}^{(1)} : y^{(1)})$, F_1^1 is a fuzzy set whose MF is centered at $x_1^{(1)}$, F_2^1 is a fuzzy set whose MF is centered at $x_2^{(1)}, \ldots, F_p^1$ is a fuzzy set whose MF is centered at $x_p^{(1)}$, and G^1 is a fuzzy set whose MF is centered at $y^{(1)}$.
2. R^2: IF x_1 is F_1^2 and x_2 is F_2^2 and … and x_p is F_p^2, THEN y is G^2—In this rule, which is obtained from $(\mathbf{x}^{(2)} : y^{(2)})$, F_1^2 is a fuzzy set whose MF is centered at $x_1^{(2)}$, F_2^2 is a fuzzy set whose MF is centered at $x_2^{(2)}, \ldots, F_p^2$ is a fuzzy set whose MF is centered $x_p^{(2)}$, and G^2 is a fuzzy set whose MF is centered at $y^{(2)}$.

•••

R^N: IF x_1 is F_1^N and x_2 is F_2^N and $\cdots$ and x_p is F_p^N, THEN y is G^N—In this rule, which is obtained from $(\mathbf{x}^{(N)} : y^{(N)})$, F_1^N is a fuzzy set whose MF is centered at $x_1^{(N)}$, F_2^N is a fuzzy set whose MF is centered at $x_2^{(N)}, \ldots, F_p^N$ is a fuzzy set whose MF is centered at $x_p^{(N)}$, and G^N is a fuzzy set whose MF is centered at $y^{(N)}$.

This first approach to obtaining rules from numerical data is rather crude, because there will be one rule for each training pair, so that $M = N$. The centers of the antecedent and consequent MFs in each rule are completely determined by the data pair that is used to create the rule. Usually, all other MF parameters are specified ahead of time by the designer (e.g., standard deviations of Gaussian MFs).

Very closely related to this method are clustering methods for obtaining rules from data. These methods obtain fewer than N rules because they find one rule for a cluster of training data [see, e.g., Chiu (1994, 1997a, b), Sugeno and Yasukawa (1993) and Yager and Filev (1994)].

4.2.1.2 WM Method

This method (Wang and Mendel 1992c), also known as the *Wang–Mendel Method* (Cox 1995), leads to a fuzzy system with $M \le N$ rules and is very widely used because it is simple.[2] Given a set of data pairs (also called *input–output pairs*) $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots, (\mathbf{x}^{(N)} : y^{(N)})$, its steps are:

1. Let $[X_1^-, X_1^+], [X_2^-, X_2^+], \ldots, [X_p^-, X_p^+], [Y^-, Y^+]$ be the domain intervals of the input and output variables, respectively, where domain interval implies the interval the variable is most likely to lie in. Each domain interval is divided into $2L + 1$ regions, where L can be different for each variable. Then, MFs are assigned to the regions, labeled *SL* (Small L), …, *S1*-(Small 1), *CE* (Center), *B1*-(Big 1), …, and *BL* (Big L). Of course, other label names can be used instead of these names, and the choices of their MFs should be related to a careful understanding of the application.
2. Because of overlapping MFs, it frequently happens that $x_k^{(t)}$ is in more than one fuzzy set. The membership of each input–output point is therefore evaluated in regions where it may occur, and $x_1^{(t)}, x_2^{(t)}, \ldots, x_p^{(t)}$, or $y^{(t)}$ is assigned to the region with maximum membership.
3. To resolve conflicting rules, i.e., rules with the same antecedent MFs and different consequent MFs, a degree is assigned to each rule as follows. Let $\mu_{X_k}(x_k^{(t)})$ denote the membership of the kth input variable in the region $[X_k^-, X_k^+]$ with maximum membership, and let $\mu_Y(y^{(t)})$ denote the membership of the output variable in the region $[Y^-, Y^+]$ with maximum membership, where X_k and Y are labels from their corresponding sets *SL*, …, *S1*, *CE*, *B1*, …, *BL*. Then, the *degree for the* tth *rule* R^t, $D(R^t)$, is defined as $(t = 1, \ldots, N)$:

$$D(R^t) = \left[\prod_{k=1}^{p} \mu_{X_k}\left(x_k^{(t)}\right)\right] \times \mu_Y(y^{(t)}) \tag{4.9}$$

4. In the event of conflicting rules, the rule with the highest degree, $D(R^t)$, is kept in the rule-base, and all other conflicting rules are discarded.

Knowing the antecedents and the consequent for all rules, as well as their MFs, it is straightforward to translate this information into a FBF expansion for the WM

[2]As of February 14, 2017, according to Google Scholar, Wang and Mendel (1992c) had more than 3000 citations.

rules, as in (4.1). Note that all of the parameters of this fuzzy system have been specified during this one-pass design.

This method can be applied to noisy or noise-free data. If it is applied to noisy data the one should use a non-singleton fuzzifier for which one must determine σ_{X_k} $(k = 1, \ldots, p)$. This requires the variance (or a range on the variance) of the additive measurement noise on each measured input be known ahead of time (which is not very likely), or that it (they) can be estimated from the data; otherwise, one must also use another design method in which the variance parameters are tuned.

Although the WM Method is simple, it does have some shortcomings, namely: (1) how to choose the parameters of the antecedent and consequent MFs is left as an open issue, and (2) it can lead to a fuzzy system that has too many rules.

Example 4.4 Figure 4.1 depicts a time series that will be forecast using a fuzzy system whose Zadeh rules have five antecedents, $x_1^{(t)}$, $x_2^{(t)}$, ..., and $x_5^{(t)}$. The forecasted value of $x(k)$ is called $x_6^{(t)}$. Variable $x(k)$ has been assigned seven linguistic terms $(S3, S2, S1, CE, B1, B2, B3)$. Normal trapezoidal MFs have been chosen for $S3$ and $B3$, and normal isosceles triangle MFs have been chosen for $S2$, $S1$, CE, $B1$ and $B2$.

From the locations of $x_1^{(t)}, x_2^{(t)}, \ldots, x_5^{(t)}$ and $x_6^{(t)}$ that are shown on Fig. 4.1, observe (approximately) that $x_1^{(t)}$ has degree 0.45 in $B2$ and 0.75 in $B1$, $x_2^{(t)}$ has degree 0.2 in $S1$ and 0.75 in $S2$, $x_3^{(t)}$ has degree 0.45 in $S2$ and 0.6 in $S3$, $x_4^{(t)}$ has degree 0.4 in $S1$ and 0.75 in CE, $x_5^{(t)}$ has degree 1.0 in $S1$ and 0.2 in CE, and $x_6^{(t)}$ has

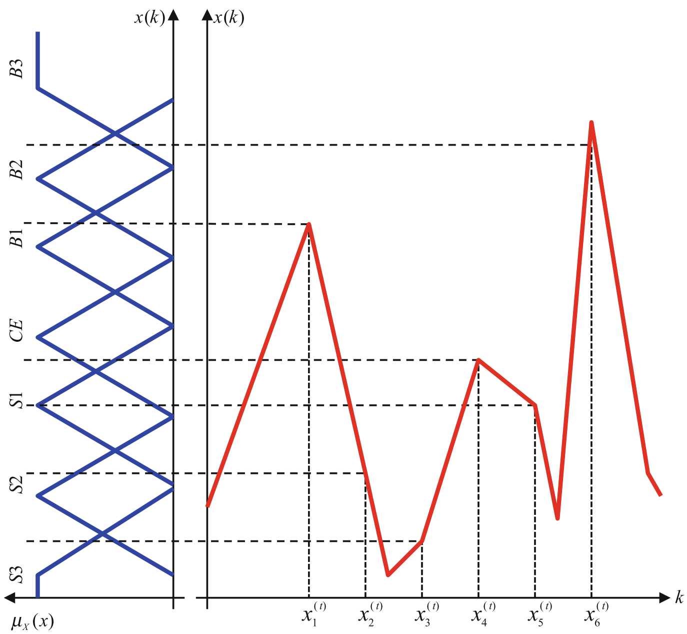


Fig. 4.1 Time series and MFs for Example 4.4

degree 0.3 in $B3$ and 0.6 in $B2$. Assigning each variable to the region with maximum degree, $x_1^{(t)}$ is considered to be $B1$, $x_2^{(t)}$ is considered to be $S2$, $x_3^{(t)}$ is considered to be $S3$, $x_4^{(t)}$ is considered to be CE, $x_5^{(t)}$ is considered to be $S1$, and $x_6^{(t)}$ is considered to be $B2$. Consequently, the Zadeh rule for this set of data is:

$$R_Z^{(t)}: \text{IF } x_1^{(t)} \text{ is } B1 \text{ and } x_2^{(t)} \text{ is } S2 \text{ and } x_3^{(t)} \text{ is } S3 \text{ and } x_4^{(t)} \text{ is CE and } x_5^{(t)} \text{ is } S1, \text{ THEN } y^{(t)} \text{ is } B2$$

The degree that is assigned to this rule (in case of a rule conflict) is:

$$\begin{aligned} D(R^t) &\equiv \mu_X\left(x_1^{(t)}\right) \times \mu_X\left(x_2^{(t)}\right) \times \cdots \times \mu_X\left(x_5^{(t)}\right) \times \mu_X(y^{(t)}) \\ &= 0.75 \times 0.75 \times 0.60 \times 0.75 \times 1.0 \times 0.60 = 0.1519 \end{aligned}$$

Wang (2003) also introduced two other rule-extracting methods (left to the reader) that are variations of the WM method and are used to solve different problems for different purposes. More specifically, he shows how to:

1. Extract specific rules targeted at a particular region and rules with different resolutions with flexible choices of the MFs.
2. Extrapolate the rules over regions not covered by the data.
3. Rank the importance of the input variables based on fuzzy predictive models (this will help to select the most relevant factors that influence the predicted variable).
4. Fine-tune and prune the fuzzy predictive models.

4.2.2 Least Squares Method

In the least squares method [e.g., Wang (1992, 1994), Wang and Mendel (1992a)] all of the antecedent parameters that are in the FBFs are fixed ahead of time by the designer. In a Mamdani fuzzy system, only the centers of the consequent MFs, the λ^l in (4.1), are tuned. In a TSK fuzzy system, only the rule consequent parameters, the c_i^l in (4.3), are tuned. The number of FBFs (i.e., the number of rules), M, must also be specified subject to the design constraints that are given in (4.8). Then, a standard least-squares optimization problem can be formulated [e.g., Mendel (1995b)], leading to a straightforward least-squares solution for λ^l or c_i^l.

Example 4.5 Here the least squares formulation and its solution are illustrated for a Mamdani fuzzy system. Beginning with (4.1), but using the notation for the elements in the data pairs, one has $(t = 1, \ldots, N)$:

$$y(\mathbf{x}^{(t)}) = \sum_{l=1}^{M} \lambda^l \phi_l(\mathbf{x}^{(t)}) \tag{4.10}$$

Collecting these N equations, they can be expressed in vector-matrix format, as:

$$\mathbf{y} = \Phi\theta \tag{4.11}$$

where

$$\mathbf{y} = \left[y^{(1)}, \ldots, y^{(N)}\right]^{\mathrm{T}} = \left[y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)})\right]^{\mathrm{T}} \tag{4.12}$$

$$\mathbf{x} = [x_1, x_2, \ldots, x_p]^{\mathrm{T}} \tag{4.13}$$

$$\theta = [\lambda^1, \ldots, \lambda^M]^{\mathrm{T}} \tag{4.14}$$

$$\Phi = \begin{pmatrix} \phi_1(\mathbf{x}^{(1)}) & \cdots & \phi_M(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}^{(N)}) & \cdots & \phi_M(\mathbf{x}^{(N)}) \end{pmatrix} \tag{4.15}$$

Matrix Φ, is called a *fuzzy basis function matrix*, and its components, $\phi_l(\mathbf{x})$ $(l = 1, \ldots, M)$, are defined in (4.2).

The least squares design for θ, θ_{LS}, is obtained by minimizing

$$J(\theta) = \frac{1}{2}[\mathbf{y} - \Phi\theta]^T[\mathbf{y} - \Phi\theta] \tag{4.16}$$

with respect to θ. Doing this minimizes the sum of the squared errors between the left- and right-hand sides of (4.10) for all $i = 1, \ldots, N$. The solution to this optimization problem can be expressed as

$$[\Phi^T\Phi]\theta_{\mathrm{LS}} = \Phi^T\mathbf{y} \tag{4.17}$$

(4.17) is a linear system of equations (known as the *normal equations*) that has to be solved for θ_{LS}. The most numerically sound methods for doing this involve orthogonal transformations [e.g., Golub and Van Loan (1983)]. Another numerically sound method for finding θ_{LS} is the singular-value decomposition (SVD). It does not solve (4.17) directly; instead, it is based on the SVD of Φ [e.g., see Lesson 4 in Mendel (1995b)].

Example 4.6 Here the least-squares formulation and its solution are illustrated for a TSK fuzzy system. Beginning with (4.3), but using the notation for the elements in the data pairs, one has $(t = 1, \ldots, N)$:

$$y_{\mathrm{TSK}}(\mathbf{x}^{(t)}) = \sum_{l=1}^{M}\sum_{j=0}^{p} c_j^l \phi_j^l(\mathbf{x}^{(t)}) \tag{4.18}$$

or, in expanded form:

$$\begin{aligned} y_{\mathrm{TSK}}(\mathbf{x}^{(t)}) &= \phi_0^1(\mathbf{x}^{(t)})c_0^1 + \phi_1^1(\mathbf{x}^{(t)})c_1^1 + \phi_2^1(\mathbf{x}^{(t)})c_2^1 + \cdots + \phi_p^1(\mathbf{x}^{(t)})c_p^1 + \cdots \\ &\quad + \phi_0^M(\mathbf{x}^{(t)})c_0^M + \phi_1^M(\mathbf{x}^{(t)})c_1^M + \phi_2^M(\mathbf{x}^{(t)})c_2^M + \cdots + \phi_p^M(\mathbf{x}^{(t)})c_p^M \end{aligned} \tag{4.19}$$

Equation (4.19) can be written more compactly as $(t = 1, \ldots, N)$

$$y_{\mathrm{TSK}}(\mathbf{x}^{(t)}) = \mathbf{g}^T(\mathbf{x}^{(t)})\mathbf{c} \quad t = 1, \ldots, N \tag{4.20}$$

where $\mathbf{g}(\mathbf{x}^{(t)})$ and $\mathbf{c}$ are $(p+1)M \times 1$ vectors, i.e.

$$\begin{cases} \mathbf{g}(\mathbf{x}^{(t)}) = [\phi_0^1(\mathbf{x}^{(t)})\ldots\phi_p^1(\mathbf{x}^{(t)})\ldots\phi_0^M(\mathbf{x}^{(t)})\ldots\phi_p^M(\mathbf{x}^{(t)})]^T \\ \qquad \mathbf{c} = [c_0^1\ldots c_p^1\ldots c_0^M\ldots c_p^M]^T \end{cases} \tag{4.21}$$

Collecting the N equations in (4.20), they can be expressed in vector-matrix format as:

$$\mathbf{y}_{\mathrm{TSK}} = \mathbf{G}\mathbf{c} \tag{4.22}$$

where the structures of $\mathbf{y}_{\mathrm{TSK}}$ and $\mathbf{G}$ are easily deduced from (4.21). Note that $\mathbf{G}$ is an $N \times (p+1)M$ matrix, where[3] $N > (p+1)M$, i.e., it has more rows than columns. This means that (4.22) is an overdetermined system of equations.

The least squares design for $\mathbf{c}$, $\mathbf{c}_{\mathrm{LS}}$, is obtained by minimizing

$$J(\mathbf{c}) = \frac{1}{2}[\mathbf{y}_{\mathrm{TSK}} - \mathbf{G}\mathbf{c}]^{\mathrm{T}}[\mathbf{y}_{\mathrm{TSK}} - \mathbf{G}\mathbf{c}] \tag{4.23}$$

with respect to $\mathbf{c}$. The solution to this optimization problem can be expressed as

$$[\mathbf{G}^T\mathbf{G}]\mathbf{c}_{\mathrm{LS}} = \mathbf{G}^T\mathbf{y}_{\mathrm{TSK}} \tag{4.24}$$

which is a linear system of $(p+1)M$ equations that has to be solved for $\mathbf{c}_{\mathrm{LS}}$. See the end of Example 4.5 for some discussions on how to do this.

Some drawbacks to the least squares method are: (1) how to choose the parameters of the antecedent MFs is left as an open issue, and (2) how to choose the number of FBFs, M, is also left as an open issue. Usually, each variable is broken up into "enough" fuzzy sets so as to cover its domain interval with "enough" resolution. The centers of the Gaussian MFs can be located at the centers of the intervals associated with each variable's fuzzy sets. The standard deviations can be chosen so that the MFs have "sufficient" overlap.

[3]In Table 4.5, when only the consequent parameters of a TSK fuzzy system are design parameters, there will be $(p+1)M$ design parameters; hence, the third line of (4.7) becomes $N > (p+1)M$.

An orthogonal least squares (OLS) procedure can be used to select the most significant FBFs (i.e., for rule reduction). The detailed formulas for the OLS procedure can be found in Wang (1994) and Wang and Mendel (1992a), and are based on the works of Chen et al. (1989, 1991). Linguistic information can be incorporated as a subset of the FBFs. The OLS procedure then establishes the simultaneous significance of linguistically and data-based FBFs. It requires choosing $M = 1, 2, \ldots, M_{\max}$ and solving a least squares optimization design problem for each value of M. The design stops when a value for M has been reached—say M^*, where $M^* \leq M_{\max}$—for which there is no longer an appreciable decline in the value of the least squares objective function.

4.2.3 Derivative-Based Methods

Two of the most popular and widely used derivative-based optimization algorithms are steepest descent and Marquardt—Levenberg. When they are used, none of the antecedent or consequent parameters are fixed ahead of time. Both algorithms need the first derivative of a mathematical objective function with respect to each MF parameter. In this section, the focus is on a steepest descent algorithm for a singleton Mamdani fuzzy system and product t-norm, for which (4.1), that uses the first line of (4.2), can be expressed as:

$$y(\mathbf{x}^{(t)}) = \sum_{l=1}^{M} \lambda^l \phi_l(\mathbf{x}^{(t)}) = \frac{\sum_{l=1}^{M} \lambda^l \prod_{i=1}^{p} \exp\left\{ -\frac{1}{2}\left[\left(x_i^{(t)} - m_{F_i^l} \right) / \sigma_{F_i^l} \right]^2 \right\}}{\sum_{l=1}^{M} \prod_{i=1}^{p} \exp\left\{ -\frac{1}{2}\left[\left(x_i^{(t)} - m_{F_i^l} \right) / \sigma_{F_i^l} \right]^2 \right\}} \tag{4.25}$$

Given the data pairs $\{(\mathbf{x}^{(t)} : y^{(t)})\}_{t=1}^{N}$, the goal here is to design this fuzzy system such that the following error function is minimized $(t = 1, \ldots, N)$:

$$J(\boldsymbol{\theta}) = \mathrm{e}^{(t)} = \frac{1}{2}\left[y(\mathbf{x}^{(t)}) - y^{(t)} \right]^2 \tag{4.26}$$

It is evident from (4.25) that y is completely characterized by λ^l, $m_{F_i^l}$ and $\sigma_{F_i^l}$ $(l = 1, \ldots, M;\ i = 1, \ldots, p)$. For clarity, instead of putting all of the unknown parameters into one vector of all such parameters, the parameters are considered below one at a time, i.e. $\boldsymbol{\theta} = \theta = \lambda^l$, or $m_{F_i^l}$ or $\sigma_{F_i^l}$.

A steepest descent optimization algorithm that only updates the parameter θ one time for each $\mathrm{e}^{(t)}$ has the following generic structure:

$$\theta(t+1) = \theta(t) - \beta_\theta \frac{\partial J(\theta)}{\partial \theta} \tag{4.27}$$

where, regardless of what θ is:

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}\left\{\frac{1}{2}\left[y(\mathbf{x}^{(t)}) - y^{(t)}\right]^2\right\} = \left[y(\mathbf{x}^{(t)}) - y^{(t)}\right]\frac{\partial}{\partial \theta}y(\mathbf{x}^{(t)}) \tag{4.28}$$

so

$$\theta(t+1) = \theta(t) + \beta_\theta\left[y(\mathbf{x}^{(t)}) - y^{(t)}\right]\frac{\partial}{\partial \theta}y(\mathbf{x}^{(t)}) \tag{4.29}$$

Example 4.7 In this example $\partial y(\mathbf{x}^{(t)})/\partial\theta$ is computed for $\theta = \lambda^l$, $m_{F_i^l}$ and $\sigma_{F_i^l}$.

(a) $\theta = \lambda^l$: In this case, it is very easy to compute $\partial y(\mathbf{x}^{(t)})/\partial\lambda^l$ from the first term on the right-hand side of (4.25), as $\partial y(\mathbf{x}^{(t)})/\partial\lambda^l = \varphi_l(\mathbf{x}^{(t)})$; hence, a steepest descent algorithm for λ^l is $(t = 1, 2, \ldots)$:

$$\lambda^l(t+1) = \lambda^l(t) - \beta_\lambda\left[y(\mathbf{x}^{(t)}) - y^{(t)}\right]\varphi_l(\mathbf{x}^{(t)}) \tag{4.30}$$

(b) $\theta = m_{F_i^l}$: Computing $\partial y(\mathbf{x}^{(t)})/m_{F_i^l}$ is much more challenging than computing $\partial y(\mathbf{x}^{(t)})/\partial\lambda^l$, because $m_{F_i^l}$ appears very non-linearly in (4.25) and is in both its numerator and denominator. To begin, let y in (4.25) be expressed as $y \equiv h/g$, where $h = \sum_{l=1}^M \lambda^l w^l$, $g = \sum_{l=1}^M w^l$ and $w^l = \prod_{i=1}^p \exp\{-\frac{1}{2}[(x_i^{(t)} - m_{F_i^l})/\sigma_{F_i^l}]^2\}$, so that $\phi_l = w^l/g$. Using the chain rule, one finds:

$$\frac{\partial y}{\partial m_{F_i^l}} = \frac{\partial y}{\partial w^l}\frac{\partial w^l}{\partial m_{F_i^l}} \tag{4.31}$$

$$\frac{\partial y}{\partial w^l} = \frac{g\frac{\partial h}{\partial w^l} - h\frac{\partial g}{\partial w^l}}{g^2} = \frac{g\lambda^l - h}{g^2} = \frac{\lambda^l - h/g}{g} = \frac{\lambda^l - y}{g} \tag{4.32}$$

$$\begin{aligned}
\frac{\partial w^l}{\partial m_{F_i^l}} &= \frac{\partial}{\partial m_{F_i^l}}\prod_{j=1}^p \exp\left\{-\frac{1}{2}\left[\left(x_j^{(t)} - m_{F_j^l}\right)/\sigma_{F_j^l}\right]^2\right\} \\
&= \frac{\partial}{\partial m_{F_i^l}}\exp\left\{-\frac{1}{2}\left[\left(x_j^{(t)} - m_{F_j^l}\right)/\sigma_{F_j^l}\right]^2\right\} \times \prod_{\substack{j=1 \\ j\neq i}}^p \exp\left\{-\frac{1}{2}\left[\left(x_j^{(t)} - m_{F_j^l}\right)/\sigma_{F_j^l}\right]^2\right\} \\
&= \prod_{j=1}^p \exp\left\{-\frac{1}{2}\left[\left(x_j^{(t)} - m_{F_j^l}\right)/\sigma_{F_j^l}\right]^2\right\} \times \frac{\left(x_i^{(i)} - m_{F_i^l}\right)}{\sigma_{F_i^l}^2} = \frac{\left(x_i^{(i)} - m_{F_i^l}\right)}{\sigma_{F_i^l}^2} \times w^l
\end{aligned} \tag{4.33}$$

Substituting (4.32) and (4.33) into (4.31), one finds that:

$$\frac{\partial y}{\partial m_{F_i^l}} = \frac{\lambda^l - y}{g} \times \frac{\left(x_i^{(i)} - m_{F_i^l}\right)}{\sigma^2_{F_i^l}} \times w^l = (\lambda^l - y_s) \times \frac{\left(x_i^{(i)} - m_{F_i^l}\right)}{\sigma^2_{F_i^l}} \times \varphi_l \quad (4.34)$$

One therefore obtains the following steepest descent algorithm for updating $m_{F_i^l}$ $(t = 1, 2, \ldots)$:

$$\begin{aligned} m_{F_i^l}(t+1) &= m_{F_i^l}(t) - \beta_m \left[y(\mathbf{x}^{(t)}) - y^{(t)}\right] \frac{\partial y}{\partial m_{F_i^l}}\bigg|_t \\ &= m_{F_i^l}(t) - \beta_m \left[y(\mathbf{x}^{(t)}) - y^{(t)}\right] \times \left[\lambda^l(t) - y(\mathbf{x}^{(t)})\right] \times \frac{\left(x_i^{(t)} - m_{F_i^l}(t)\right)}{\sigma^2_{F_i^l}(t)} \times \varphi_l(\mathbf{x}^{(t)}) \end{aligned} \quad (4.35)$$

(c) $\theta = \sigma_{F_i^l}$: Instead of updating $\sigma^2_{F_i^l}$, $\sigma_{F_i^l}$ is updated, because $\sigma^2_{F_i^l}$ must be positive (i.e., $\sigma^2_{F_i^l}$ is constrained) whereas $\sigma_{F_i^l}$ can be positive or negative (i.e., $\sigma_{F_i^l}$ is unconstrained). Because the computation of $\partial y(\mathbf{x}^{(t)})/\sigma_{F_i^l}$ is so similar to the computation of $\partial y(\mathbf{x}^{(t)})/m_{F_i^l}$, it is left as an exercise for the reader (Exercise 4.8).

Values for $\lambda^l(1)$, $m_{F_i^l}(1)$ (and $\sigma_{F_i^l}(1)$) must be provided to initialize (4.30) and (4.35). How to do this is application-dependent and is discussed in Sect. 4.3.3. Because λ^l, $m_{F_k^l}$, and $\sigma_{F_k^l}$ are parameters associated with MFs for physically meaningful quantities, it is usually possible to obtain very good initial values for them. About the worst way to initialize these parameters is to choose them randomly, because doing this will cause a derivative-based algorithm to converge very slowly. Choosing them smartly will cause this algorithm to converge much faster [see Chu and Mendel (1994)].

The learning parameters β_m, β_λ, and β_σ must also be chosen with some care. Frequently, they are chosen to be the same, say β. Choosing too large a value for β can cause the algorithm not to converge, whereas choosing too small a value for β can cause the algorithm to take a very long time to converge. In practice, one often must develop a schedule for how to choose β. One schedule is to choose it larger for the early iterations of the algorithm and to then let it become smaller for later iterations of the algorithm. Arabshahi et al. (1996) formalize some Zadeh rules for adjusting β. These rules are then turned into mathematical formulas for a Mamdani fuzzy system. This is a very clever way to adjust (adapt) β in a nonlinear manner (see Exercise 3.18).

Let one *epoch* be defined as the collection of N training data. A steepest descent algorithm has just been described in which each element of the training set is used only one time, and the fuzzy system's parameters are updated using the error function (4.26) that depends only on one data point at a time. Training occurs for only one epoch. Other variations are possible, including:

1. Apply (4.30) and (4.35) within an epoch and then for many epochs until convergence occurs.
2. Define a squared error function that depends on all N training data. Redevelop (4.30) and (4.35) for this new error function and apply the modified steepest descent algorithm until convergence occurs (see Exercise 4.6).

Recall that a feedforward neural network is a layered architecture, and that when its parameters (weights) are optimized using the method of steepest descent, the resulting algorithm is called a *backpropagation algorithm* [e.g., Haykin (1996)]. In that algorithm, the output error is propagated in a backward direction from the output layer down into lower layers, hence the name *backpropagation*. Section 3.9.1 mentioned that a fuzzy system described by (4.25) can also be viewed as a layered architecture, one with three (or more) layers. Equations (4.30) and (4.35) are therefore often referred to as *backpropagation algorithms* because of their dependence on the error $y(\mathbf{x}^{(t)}) - y^{(t)}$, which propagates from the output layer of the fuzzy system down into lower layers.

Some drawbacks to derivative-based methods are:

1. If a MF changes its mathematical formula over its domain (e.g., as would be the case for triangle and trapezoidal MFs), then derivative formulas change as well, and time-consuming domain tests have to be included.
2. They tend to only find a local extremum of an objective function rather than the global extremum, i.e., they tend to get trapped at a local extremum. Of course, there are ways to avoid getting trapped, but when derivatives are used there is a tendency to get trapped.
3. How to choose the number of FBFs, M, is left as an open issue. The next method can be used to resolve this drawback.

4.2.4 SVD-QR Method

Section 3.9.4 explained that rule explosion can be a problem for a fuzzy system, and that the singular-value decomposition (SVD) is one approach for rule reduction. The SVD of a matrix is a very powerful tool in numerical linear algebra. Among its important uses are the determination of the rank of a matrix and numerical solutions of linear least squares problems. It can be applied to square or rectangular matrices, whose elements are either real or complex. According to (Klema and Laub 1980, p. 166):

> The SVD was established for real square matrices in the 1870s by Beltrami and Jordan [see, e.g., MacDuffee (1933, p. 78)], for complex square matrices in Autonne (1902), and for general rectangular matrices in Eckart and Young (1939) (the Autonne–Eckart–Young theorem).

The SVD is presently one of the major computational tools in signal processing.

Let $\mathbf{H}$ be a $K \times M$ matrix, and $\mathbf{U}$ and $\mathbf{V}$ be two $K \times K$ and $M \times M$ unitary matrices, respectively. The SVD of $\mathbf{H}$ is:

$$\mathbf{H} = \mathbf{U}\begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{V} \tag{4.36}$$

where

$$\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \tag{4.37}$$

and

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 \tag{4.38}$$

The σ_i are the *singular values* of $\mathbf{H}$ and r is the rank of $\mathbf{H}$. This decomposition is true for both the overdetermined $(K > M)$ and underdetermined $(K < M)$ cases.

There are many excellent sources for comprehensive discussions about the SVD, including its derivation and how to compute it [e.g., Stewart (1973), Golub and Van Loan (1989), Haykin (1996) and Mendel (1995b)]. None of those details are presented here since doing so would take us too far afield from the main purpose for this section, which is to explain how the SVD can be used to choose the most important of the M FBFs in (4.1) or the $M(p+1)$ FBFs in (4.3).

The SVD provides a natural way to separate a space into dominant and subdominant subspaces. If one views the FBF matrix Φ in (4.15) as a span of the input subspace, then the SVD of Φ decomposes the span into an equivalent orthogonal span, from which one can identify the dominant and subdominant spans (Vaccaro 1991), i.e., one can identity which FBFs contribute the most to the fuzzy system, and how many of them are needed to effectively represent it. The FBFs that contribute the least can be discarded, and a reduced parsimonious fuzzy system can be designed. Note that the SVD approach only reorders the original FBFs to form a set of independent FBFs; thus, it preserves the meaning of linguistic information initially incorporated into the system.

The steps of the SVD–QR algorithm (Mouzouris and Mendel 1996, 1997; Yen and Wang 1996, 1999), stated for the FBF matrix Φ in (4.15), are:

1. Calculate the SVD of the FBF matrix Φ. Computer programs for doing this are widely available, e.g., MATLAB® (MATLAB is a registered trademark of The MathWorks, Inc.).
2. Estimate the numerical rank of Φ by examining the singular values of this matrix. A common choice for the numerical rank is the number of singular values that are above a small threshold that must be prespecified.
3. Keep that part of the SVD of Φ associated with the estimated numerical rank of Φ.

4. Using a QR algorithm [e.g., Golub and Van Loan (1983)], rank-order the FBFs that are associated with the rank-ordered singular values that are above the small threshold. Doing this leads to another FBF matrix, $\Phi_{M'}$, where

$$\Phi_{M'} = \begin{pmatrix} \phi'_1(\mathbf{x}^{(1)}) & \cdots & \phi'_{M'}(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ \phi'_1(\mathbf{x}^{(N)}) & \cdots & \phi'_{M'}(\mathbf{x}^{(N)}) \end{pmatrix} \tag{4.39}$$

in which $M' < M$ and the primed FBFs denote the fact that they have been reordered. The FBF expansion in (4.1) now is:

$$y(\mathbf{x}^{(i)}) = \sum_{l=1}^{M'} \lambda'^l \phi'_l(\mathbf{x}) \tag{4.40}$$

5. Renormalize the denominators of the M' FBFs using the firing levels for just those FBFs. If this step is not performed, then the FBFs will still be normalized by the firing levels of the original M FBFs, which defeats the purpose of the SVD–QR method [see, also Hohenson and Mendel (1994, 1996)].[4]
6. Determine the M' λ'^l parameters in (4.40) using the least squares method described earlier.

Although the SVD–QR design method provides optimal values for λ'^l in a least-squares sense, it does not provide any values for the remaining design parameters of the fuzzy system. The antecedent MF parameters must be prespecified to use the SVD-QR design method.[5] See, also, Sect. 4.2.6.

4.2.5 Derivative-Free Methods

There are now many optimization methods that do not require derivatives of mathematical objective functions. They are very attractive for tuning MF parameters in a fuzzy system, because they are a way to overcome the first two drawbacks to derivative-based methods that are listed at the end of Sect. 4.2.3, namely, having

[4]This step changes the FBF matrix that was used in Step 1; so, although it is not stated in the just mentioned references, it may be a good idea to now return to Step 1 and iterate through Steps 1–5 until no appreciable changes occur in Step 5 from one such iteration to the next.

[5]Setnes and Hellendoorn (2000) considered a pivoted QR decomposition to order and select fuzzy rules. They point out that rank-revealing methods that are based on the SVD do not produce an "importance ordering" of the rules. Our use of the SVD in the SVD–QR method does not attempt to determine the rank of a matrix. It uses a pre-specified small number to compare the singular values, accepting those above that number and rejecting the others. In that way, one is assured that the kept rules contain all the important ones, although they may not be in the order of importance.

to use domain tests so that a correct derivative formula is used, and getting trapped at a local extremum.

The derivative-free methods (algorithms) are often referred to as *evolutionary* (Simon[6] 2013) or *bio-inspired* (Castillo and Melin 2012) and include: genetic algorithms, evolutionary programming, genetic programming, simulated annealing, ant colony optimization, particle swarm optimization (PSO), quantum particle swarm optimization (QPSO), differential evolution, biogeography-based optimization, cultural, opposition-based learning, Tabu search, artificial fish swarm, shuffled leap frog, firefly, bacterial foraging, artificial bee colony, gravitational search, harmony search, etc. In theory, these methods are able to find the global extremum of an objective function, which tends to make them more powerful than derivative-based methods, but, in actual practice, one really never knows whether a solution is at the global extremum. Of course, if by repeating the optimization procedure many times one arrives at the same solution, then one's confidence that the solution is at the global extremum is increased.

At a very high level, many of these derivative-free methods can be viewed as creating a collection (swarm) of candidate solutions (particles) that communicate among each other (each method does this in its own bio-inspired way) from the present iteration to the next iteration, until a winning solution emerges. A price paid for using such derivative-free methods is massive amounts of computation in which the evaluation of the objective function for each candidate solution (particle) is usually the most intensive computation. Many of these methods also employ randomness in order to change the internal structures of their particles. If computation time is not important, then these are very powerful methods.

With the availability of such a multitude of bio-inspired algorithms, a natural question to ask is: Which method should one use to optimize the parameters of a fuzzy system? To-date, there does not seem to be a general answer to this question, and so one tends to use the bio-inspired algorithm that one is most familiar with, and then compares its results with those obtained by using some of the other methods [e.g., Castillo et al. (2012)]. In that spirit, one such algorithm—QPSO—is now explained.

QPSO [e.g., Sun et al. (2012)] is a globally convergent (Wei et al. 2010) iterative search algorithm that generally outperforms the original PSO (Kennedy and Eberhart 1995; Wang et al. 2011) and has fewer parameters to control. Each particle represents a possible solution to an optimization problem. The position of each particle is updated (in each QPSO iteration) by using its most recent own best solution, mean of the personal best positions of all (N_m) particles, and the global best solution found by all particles so far.

QPSO finds optimized $\boldsymbol{\theta}$ based on the following criterion: $\min_{\boldsymbol{\theta}_m} J(\boldsymbol{\theta}_m)$. The *current position* (vector) of the mth particle is defined, as $(m = 1, \ldots, N_m)$:

[6]This book, which is more than 700 pages, is a very scholarly treatment of these algorithms and has more than 40 pages of references. The list of these algorithms that is given next has been taken from the table of contents of this book.

$$\boldsymbol{\theta}_m = \mathrm{col}(\theta_{m,1}, \theta_{m,2}, \ldots, \theta_{m,N_\theta}) \tag{4.41}$$

where N_θ denotes the number of parameters to be optimized in a fuzzy system. A *particle best position* (i.e., the position that produces the minimal value of $J(\boldsymbol{\theta}_m)$ over the entire history of that particle), is denoted $\mathbf{p}_m = \mathrm{col}(p_{m,1}, p_{m,2}, \ldots, p_{m,N_\theta})$ and is computed as $(j = 1, \ldots, N_\theta)$:

$$p_{m,j}(t+1) = \eta p_{m,j}(t) + (1-\eta) p_{\mathrm{gbest},j}(t) \tag{4.42}$$

where $t = 1, \ldots, T-1$ is the *index of a generation* (iteration), $p_{m,j}(1)$ is initialized by $\theta_{m,j}(1)$, η is a random variable uniformly distributed in $(0, 1]$, and $\mathbf{p}_{\mathrm{gbest}}(t)$ (whose components are $p_{\mathrm{gbest},j}(t)$) denotes the *global best (gbest) position* found in the entire history of the swarm at the tth iteration, i.e.

$$\mathbf{p}_{\mathrm{gbest}}(t) = \arg_{\mathbf{p}_m(t)} \min_{\forall m=1,\ldots,N_m} J(\mathbf{p}_m(t)) \tag{4.43}$$

A global point, the *mean best position* of the population is introduced into QPSO; it is denoted $\mathbf{m}(t)$ and is defined as the sample mean of the $\mathbf{p}_m(t)$ positions of all N_m particles, i.e.,

$$\mathbf{m}(t) = \frac{1}{N_m} \sum_{m=1}^{N_m} \mathbf{p}_m(t) \tag{4.44}$$

At the end of each iteration, a new position of a particle is obtained as $(j = 1, \ldots, N_\theta)$:

$$\theta_{m,j}(t+1) = p_{m,j}(t+1) \pm \beta \left| m_j(t) - \theta_{m,j}(t) \right| \ln(1/\rho) \tag{4.45}$$

where β, called the *contraction—expansion coefficient*, is used to control the convergence speed of the algorithm [see Sun et al. (2012) for some examples that use values of β that range from 0.5 to 1.1], and ρ is also a random variable uniformly distributed in $(0, 1]$. In (4.45), the plus and minus signs are randomly selected to generate the new position of a particle.

QPSO can be applied directly to a singleton fuzzy system, where the parameters of all particles are initialized randomly.

The following two-step design procedure is advocated when QPSO is used to optimize the parameters of a non-singleton fuzzy system:

1. Design a singleton type-1 fuzzy system using QPSO, as just described.
2. Design a non-singleton fuzzy system by optimizing its parameters using QPSO in which one particle is associated with the just QPSO-optimized singleton type-1 fuzzy system, and where the parameters of all the remaining particles are initialized randomly.

Table 4.6 Pseudo-code for QPSO as used in optimal designs of type-1 fuzzy systems (Mendel 2014)

Initialize $\theta_1(1)$ either randomly (for the design of a singleton fuzzy system) or as a singleton fuzzy system particle (for the design of a non-singleton fuzzy system), and all other $\theta_m(1)$ randomly $(m = 2,\ldots,N_m)$
Set $\mathbf{p}_m(1) = \boldsymbol{\theta}_m(1)$ $(m = 1,\ldots,N_m)$
For t = 1 to G-1
Calculate $\mathbf{m}(t) = \frac{1}{N_m}\sum_{m=1}^{N_m}\mathbf{p}_m(t)$
Calculate $J(\mathbf{p}_m(t))$ $(m = 1,\ldots,N_m)$
Calculate $\mathbf{p}_{\text{gbest}}(t) = \arg\min_{\mathbf{p}_m(t),\ \forall m=1,\ldots,N_m} J(\mathbf{p}_m(t))$
for m = 1 to N_m (number of particles)
Calculate $J(\boldsymbol{\theta}_m(t))$
If $J(\boldsymbol{\theta}_m(t)) < J(\mathbf{p}_m(t))$ $\mathbf{p}_m(t) = \boldsymbol{\theta}_m(t)$ **end if**
for j = 1 to N_θ (number of components in each particle)

$\eta = \text{rand}(0,1)$
 $p_{m,j}(t+1) = \eta \times p_{m,j}(t) + (1-\eta)\times p_{gbest,j}(t)$
 $\rho = \text{rand}(0,1)$
 if $\text{rand}(0,1) > 0.5$ **then**
 $\theta_{m,j}(t+1) = p_{m,j}(t+1) - \beta \mid m_j(t) - \theta_{m,j}(t)\mid \ln(1/\rho)$
 else
 $\theta_{m,j}(t+1) = p_{m,j}(t+1) + \beta \mid m_j(t) - \theta_{m,j}(t)\mid \ln(1/\rho)$
 end if
 end for
 end for

end for

Pseudocode for QPSO that abides by this two-step procedure is given in Table 4.6.

Theorem 4.1 *By virtue of the QPSO algorithm, the performance of the optimized non-singleton type-1 fuzzy system cannot be worse than that of the optimized singleton type-1 fuzzy system.*

Proof See Appendix 1 to this chapter.

By this two-step systematic design approach it is not possible for the performance of an optimized non-singleton type-1 fuzzy system to be worse than that of an optimized singleton type-1 fuzzy system.[7] This does not mean that the performance of the optimized non-singleton type-1 fuzzy system will be *significantly better* than that of the optimized singleton type-1 fuzzy system. There is no analysis that is available to-date that focuses on such relative performance improvements. Of course, relative improvements are very application dependent because objective function $J(\boldsymbol{\theta})$ is application dependent.

By the way, this kind of guaranteed performance improvement can also be achieved with a derivative-based algorithm if after each iteration the performance of the non-singleton type-1 fuzzy system is compared with that obtained from a singleton type-1 fuzzy system. A new set of parameters is accepted only if the performance from the non-singleton type-1 fuzzy system is better[8] than that of the singleton type-1 fuzzy system.

Example 4.8 In Step 2 of the above design procedure, one particle is associated with the just QPSO-optimized singleton type-1 fuzzy system. Suppose, for example that the non-singleton type-1 fuzzy system is Mamdani and uses height defuzzification. The structure of a particle for such a fuzzy system is given as:

$$\boldsymbol{\phi}_{\text{NST1}} = \text{col}\left(\underbrace{\overbrace{m_1^1,\sigma_1^1}^{\text{Antecedent 1}},\ldots,\overbrace{m_p^1,\sigma_p^1}^{\text{Antecedent}\,p},\overbrace{\bar{y}^1}^{\text{Consequent}}}_{\text{Rule1}};\ldots;\underbrace{\overbrace{m_1^M,\sigma_1^M}^{\text{Antecedent 1}},\ldots,\overbrace{m_p^M,\sigma_p^M}^{\text{Antecedent}\,p},\overbrace{\bar{y}^M}^{\text{Consequent}}}_{\text{Rule}M};\underbrace{\sigma_{X_1},\sigma_{X_2},\ldots,\sigma_{X_p}}_{\text{Non-singleton fuzzifier}}\right) \tag{4.46}$$

The singleton type-1 fuzzy system particle that is used in Step 2, which must be of the same length as this non-singleton type-1 fuzzy system particle, begins with (4.46) and can be expressed as:

$$\boldsymbol{\phi}_{\text{ST1}} = \text{col}\left(\underbrace{\overbrace{m_1^1,\sigma_1^1}^{\text{Antecedent1}},\ldots,\overbrace{m_p^1,\sigma_p^1}^{\text{Antecedentp}},\overbrace{\bar{y}^1}^{\text{Consequent}}}_{\text{Rule1}};\ldots;\underbrace{\overbrace{m_1^M,\sigma_1^M}^{\text{Antecedent 1}},\ldots,\overbrace{m_p^M,\sigma_p^M}^{\text{Antecedent}\,p},\overbrace{\bar{y}^M}^{\text{Consequent}}}_{\text{Rule}M};\underbrace{0,0,\ldots,0}_{\text{Singleton fuzzifier}}\right) \tag{4.47}$$

In (4.47): (1) All of the MF parameters are taken from the QPSO-optimized singleton type-1 fuzzy system design; and, (2) By setting the value of the fuzzifier

---

[7]If a fuzzy system cannot be designed by means of optimization, but can only be designed by trial and error, then it is possible that the performance of the non-singleton type-1 fuzzy system may be worse than that of a singleton type-1 fuzzy system. This occurs because it may not be possible to try all possible combinations of the design parameters, and is therefore the result of an incomplete design procedure. One must be very cautious about drawing conclusions from such incomplete trial and error designs.

[8]How much better has to be specified by the designer.

standard deviations in (4.46) equal to 0, the non-singleton fuzzifier reduces to singleton fuzzifier. In this way, it is straightforward to embed a singleton type-1 fuzzy system particle into a non-singleton type-1 fuzzy system particle.

4.2.6 Iterative Design Methods

There are many ways to combine the optimization methods that have been described in Sects. 4.2.1–4.2.5, such as:

- By combining the SVD–QR method with a derivative-based or particle-based method, one can design all of the parameters of a fuzzy system including the number of the most significant rules, M'. The following iterative design method can be very successful:

  1. Fix the number of rules, M, at a reasonable value.
  2. Use a derivative-based or particle-based method to design all the antecedent and consequent MF parameters.
  3. Apply the SVD–QR method to the results of the derivative-based or particle-based method to determine $M' < M$ FBFs.
  4. Renormalize the FBFs and recompute the linear combining parameters using least-squares.
  5. If performance is acceptable, STOP. Otherwise, return to Step 2 for a retuning of the antecedent and consequent MF parameters.

- By applying the SVD-QR method to the FBF matrix that can be created after Zadeh rules are obtained from the WM method.
- By using an evolutionary or bio-inspired optimization method that is set up not only to optimize MF parameters, but also other things such as [e.g., Rutkowski (2004)]: which antecedents to use as well as their number (i.e., p), the number of linguistic terms for each variable (i.e., $Q_1, \ldots, Q_p$), the number of rules (i.e., M), the t-norm used (i.e., minimum or product), Mamdani product or minimum, and the defuzzification method (i.e., centroid, height or COS).

4.2.7 Remarks

Although the discussions in Sects. 4.2.3–4.2.6 have been mainly in the context of a Mamdani fuzzy system, they apply as well for a TSK fuzzy system, something that is left to the reader to explore (see Exercises 4.7, 4.10 and 4.12).

When a type-1 fuzzy system is going to be used as part of a consumer (or military) product then it should be designed to meet prespecified performance specifications instead of by optimizing a mathematical objective function (unless the performance specifications can be expressed by means of such a function).

If this can be accomplished by using a singleton type-1 fuzzy system, the design can stop, but if it cannot then the design process should proceed to a non-singleton type-1 fuzzy system. Unfortunately, such designs do not appear in the open literature because usually product performance specifications are proprietary or classified. Consequently, academics (such as the author) focus on optimal designs.

Finally, there are additional benefits to using a type-1 fuzzy system over a crisp system that may not be encapsulated by a mathematical objective function, or, to put it another way, a design that is based only on minimizing a mathematical objective function may not in the end be evaluated for its success just by examining the minimum (or maximum) value for that objective function. For example, greater smoothness can be obtained by using type-1 fuzzy sets over crisp sets, and smoothness may be difficult to quantify using an objective function.

4.3 Case Study: Forecasting of Time Series

Let $s(k)$ $(k = 1, 2, \ldots, N)$ be a time series, such as daily temperatures of London England, or hourly measurements of a stock index (e.g., Dow-Jones), or a diabetic person's insulin level. Measured-values of $s(k)$ are denoted $x(k)$, where $x(k) = s(k) + n(k)$ in which $n(k)$ denotes measurement error (e.g., noise). The problem of forecasting a time series (i.e., prediction) is:

> Given a window of p past measurements of $s(k)$, namely $x(k-p+1)$, $x(k-p+2), \ldots, x(k)$, determine an estimate of a future value of s, $\hat{s}(k+l)$, where p and l are fixed positive integers.

If the measurements are noise-free (i.e., perfect), then $x(k-p+1), x(k-p+2), \ldots, x(k)$ in this problem statement are replaced by $s(k-p+1), s(k-p+2), \ldots, s(k)$, respectively.

Forecasting is a very important problem that appears in many disciplines. Better weather forecasts can save lives in the event of a catastrophic hurricane; better financial forecasts can improve the return on an investment; better forecasts of a diabetic person's insulin level could be life saving, etc.

When $l = 1$ the single-stage forecaster of s is obtained; when $l = 2$ the two-stage forecaster of s is obtained; and, in general, for arbitrary values of l, the l-stage forecaster of s is obtained. For illustrative purposes, our attention is focused on the case of $l = 1$.

Suppose that one is given a collection of N data points, $x(1), x(2), \ldots, x(N)$. Then, as is commonly done when neural networks are used to forecast a time series, this data set is partitioned into two subsets[9] a *training* data subset with D data

[9]In some applications, the data are partitioned into three subsets: a *training* data subset with D_1 data points, a *validation* subset with D_2 data points and a *testing* subset with $N - D_1 - D_2$ data points. The validation subset is used to determine when to stop the training. The testing subset is used to evaluate the final trained system, but is not used as part of the training process.

points, $x(1), x(2), \ldots, x(D)$, and a *testing* data subset with $N - D$ data points, $x(D+1), x(D+2), \ldots, x(N)$. Because a window of p data points will be used to forecast the next data point, there are at most $D - p$ training pairs, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(D-p)}$, where

$$\begin{aligned} \mathbf{x}^{(1)} &= [x(1), x(2), \ldots, x(p), x(p+1)]^{\mathrm{T}} \\ \mathbf{x}^{(2)} &= [x(2), x(3), \ldots, x(p+1), x(p+2)]^{\mathrm{T}} \\ &\cdots \\ \mathbf{x}^{(D-p)} &= [x(D-p), x(D-p+1), \ldots, x(D-1), x(D)]^{\mathrm{T}} \end{aligned} \tag{4.48}$$

In (4.48), the first p elements of $\mathbf{x}^{(t)}$ are the inputs to the forecaster and the last element of $\mathbf{x}^{(t)}$ is the desired output of the forecaster, i.e.,[10]

$$\mathbf{x}^{(t)} = [p \times 1 \text{ input, desired output}]^{\mathrm{T}} = [x_1^{(t)}, x_2^{(t)}, \ldots, x_p^{(t)}, x_{p+1}^{(t)}]^{\mathrm{T}} \tag{4.49}$$

where $t = 1, 2, \ldots, D - p$. The training data are used in a fuzzy system forecaster to establish its rules.

Two ways to extract rules from the numerical training data have been described in Sect. 4.2.1. Because a predicted value of s will depend on p past values of x, there will be p antecedents in each rule. Let these p antecedents be denoted $x_1, x_2, \ldots, x_p$. The interesting feature of time series forecasting is that, although each rule has p antecedents, *these antecedents are all associated with the same variable*, e.g., daily temperature in London, and so is the consequent.

The forecasting rules used in this case study are Zadeh rules [as in (3.1)], and the fuzzy systems used here are Mamdani [(4.1) and (4.2)]. To fully specify these fuzzy systems, all of the items that are in Table 4.1 need to be decided upon. The following choices are common to all of the fuzzy systems that are described below: (1) rules have 4 antecedents ($p = 4$), (2) product t-norm, (3) height defuzzification, and (4) Gaussian MFs.

4.3.1 Mackey–Glass Chaotic Time Series

Today, chaos is having an impact on many different fields including physics, biology, chemistry, economics, and medicine [e.g., Casdagli (1992), Farmer (1982) and Rasband (1990)]. Very briefly, chaotic behavior can be described as bounded fluctuations of the output of a *non linear* system with high degree of sensitivity to *initial conditions* (Casdagli 1992), i.e., trajectories with nearly identical initial

---

[10]In Sects. 4.1 and 4.2, the data pairs were designated as $(\mathbf{x}^{(1)}; y^{(1)}), \ldots, (\mathbf{x}^{(N)}; y^{(N)})$. In (4.49) $[x_1^{(t)}, \ldots, x_p^{(t)}]^{\mathrm{T}}$ plays the role of $\mathbf{x}^{(t)}$, $x_{p+1}^{(t)}$ plays the role of $y^{(t)}$ and $D - p$ plays the role of N.

conditions can differ a lot from each other. A system exhibiting chaotic dynamics evolves in a *deterministic* manner; however, the correlation of observations from such a system appears to be limited, so the observations appear to be uncorrelated; thus, forecasting for such a system is particularly difficult (Rasband 1990).

Mackey and Glass (1977) is an important paper in which the authors "associate the onset of disease with bifurcations in the dynamics of first-order differential-delay equations which model physiological systems". Equation (4b) of that paper, which has become known as the *Mackey–Glass equation*, is a nonlinear delay differential equation, one form of which is

$$\frac{\mathrm{d}s(t)}{\mathrm{d}t} = \frac{0.2s(t-\tau)}{1+s^{10}(t-\tau)} - 0.1s(t) \tag{4.50}$$

For $\tau > 17$ (4.50) is known to exhibit chaos.

To demonstrate the qualitative nature of the Mackey–Glass equation, representative portions of the associated Mackey–Glass time series [i.e., the solution of (4.50)] are displayed for two values of τ in Fig. 4.2a, b. The corresponding two-dimensional phase plots are depicted in Fig. 4.2c, d. From these plots one is able to distinguish periodic behavior for small values of τ and chaotic behavior for larger values of τ.

The Mackey–Glass time series (for $\tau > 17$) has become one of the benchmark problems for time-series prediction in both the neural network and fuzzy logic fields [e.g., Lapedus and Farber (1987), Moody (1989), Moody and Darkin (1989), Sanger (1991), Wang (1994), Hohenson and Mendel (1996) and Jang et al. (1997)]. This case study is directed at single-stage forecasting for a Mackey–Glass time series in which $\tau = 30$.

As was mentioned above, the single-stage forecasters, using different kinds of fuzzy systems, are based on $D - p$ training pairs. These training pairs were obtained by simulating (4.50) for $\tau = 30$. This was done by converting (4.50) to a discrete-time equation, using Euler's method with a step size equal to 1 (Quinney 1985). Because of the 30 time-unit delay, the resulting discrete-time equation requires 31 initial values. These first 31 values of $s(k)$ (i.e., $s(1), s(2), \ldots, s(31)$) were chosen randomly a number of times until a time series was obtained that looked interesting. Thereafter, the same 31 initial values were used to produce a deterministic (albeit, chaotic) time series that was used in the forecasting studies that are described below and also in Chap. 10. The noise-free Mackey–Glass time series used in our study $[s(1001), s(1002), \ldots, s(2000)]$ is depicted in Fig. 4.3.

4.3.2 One-Pass Design: Singleton Fuzzification

To begin, the first 504 data, $s(1001), s(1002), \ldots, s(1504)$, were used to design a one-pass fuzzy system forecaster, and the remaining 496 data, $s(1505), s(1006), \ldots, s(2000)$, were used for testing the design. Only the four

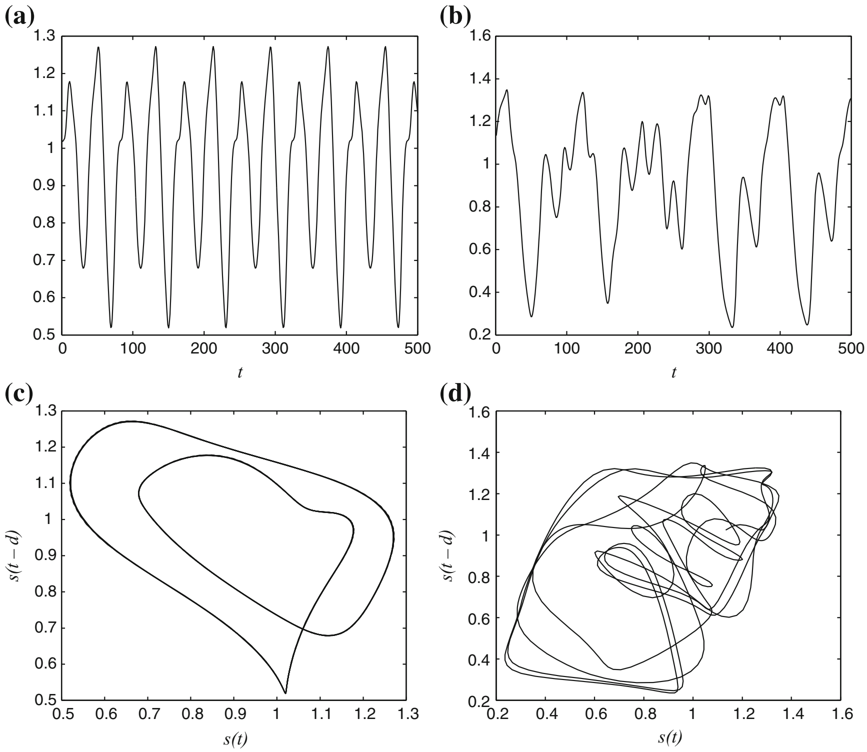


Fig. 4.2 **a** and **b** are representative samples of the Mackey–Glass time series after letting transients relax. **c** and **d** are the corresponding phase plots of the time segments depicted in (**a**) and (**b**). Note that "d" in $s(t-d)$ on the vertical axis denotes the delay (τ) used in the Mackey–Glass equation; it is 13 for (**a**) and (**c**) and 30 for (**b**) and (**d**)

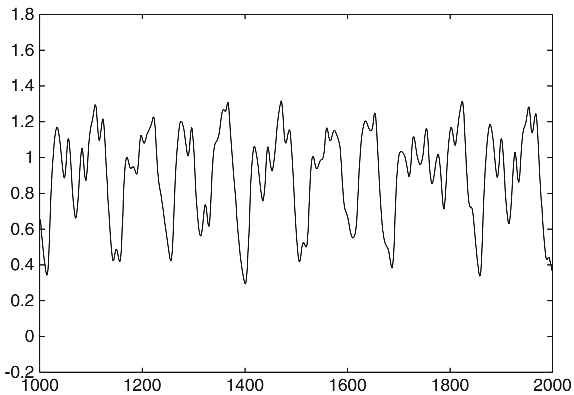


Fig. 4.3 Mackey–Glass time series for noise-free data, $s(1001), s(1002), \ldots, s(2000)$

antecedents $s(k-3)$, $s(k-2)$, $s(k-1)$ and $s(k)$ were used to predict $s(k+1)$. The performance of this design was evaluated using the following root mean-squared error (RMSE):

$$\mathrm{RMSE}_s = \sqrt{\frac{1}{496}\sum_{k=1504}^{1999}\left[s(k+1) - y_s(\mathbf{s}^{(k)})\right]^2} \tag{4.51}$$

where $\mathbf{s}^{(k)} = [s(k-3), s(k-2), s(k-1), s(k)]^{\mathrm{T}}$, $y_s(\mathbf{s}^{(k)})$ is found by substituting the top line of (4.2) into (4.1), and the subscript "s" on RMSE reminds us that this is a singleton design.

The Sect. 4.2.1.1 one-pass data assignment method was used to construct 500 Zadeh rules from $s(1001), s(1002), \ldots, s(1504)$. The antecedent and consequent MFs were centered at the noise-free measurements in each one of the 500 rules, and the standard deviation of the Gaussians was set equal to 0.1. This established the Mamdani singleton fuzzy system forecaster $y_s(\mathbf{s}^{(k)})$. Then $s(1505), s(1506), \ldots, s(2000)$, were used to compute the one-pass (OP) RMSE in (4.51) as $\mathrm{RMSE}_s(\mathrm{OP}) = 0.0438$.

Although this design is very simple, it leads to a fuzzy system with 500 rules, which is very excessive. The next design demonstrates better RMSE performance using only 16 rules.

4.3.3 Derivative-Based Design: Singleton Fuzzification

Next, as in Jang (1993), only two fuzzy sets were used (which is as small a number as one can use in a fuzzy system) for each of the four antecedents (their names are unimportant because the interest here is only in a numerical forecast); hence, the number of rules is very small and equals $2^4 = 16$. Each rule is characterized by eight antecedent MF parameters (the mean and standard deviation for each of the four Gaussian MFs) and one consequent parameter, $\bar{y}$. The initial location of each Gaussian antecedent MF was based on the mean, m_s, and the standard deviation, σ_s, of the data in the 504 training samples, $s(1001)$, $s(1002), \ldots, s(1504)$. More specifically, the means of each and every antecedent's two Gaussian MFs were initially chosen as $m_s - 2\sigma_s = 0.3793$ or $m_s + 2\sigma_s = 1.4272$, respectively, and their standard deviations were initially chosen as $2\sigma_s$. The center of each consequent's MF, $\bar{y}\,(i = 1, \ldots, 16)$, was initially chosen to be a random number from the interval $[0, 1]$. After training using a steepest descent algorithm, as described in Sect. 4.2.3, in which the learning parameter $\beta_\theta = 0.2$ (this same value was used for all of the parameters), the fuzzy system forecaster was fixed. Its performance was then evaluated using the RMSE in (4.51).

Table 4.7 Initial values for the centers of the Gaussian antecedent MFs and the center of the consequent set

Rule No.	Initial value for centers of the four antecedent MFs				Initial value for $\bar{y}^l$
1	0.3793	0.3793	0.3793	0.3793	0.5314
2	0.3793	0.3793	0.3793	1.4272	0.3831
3	0.3793	0.3793	1.4272	0.3793	0.0159
4	0.3793	0.3793	1.4272	1.4272	0.8181
5	0.3793	1.4272	0.3793	0.3793	0.6931
6	0.3793	1.4272	0.3793	1.4272	0.1209
7	0.3793	1.4272	1.4272	0.3793	0.4647
8	0.3793	1.4272	1.4272	1.4272	0.9975
9	1.4272	0.3793	0.3793	0.3793	0.9522
10	1.4272	0.3793	0.3793	1.4272	0.6991
11	1.4272	0.3793	1.4272	0.3793	0.2673
12	1.4272	0.3793	1.4272	1.4272	0.7625
13	1.4272	1.4272	0.3793	0.3793	0.6460
14	1.4272	1.4272	0.3793	1.4272	0.6483
15	1.4272	1.4272	1.4272	0.3793	0.3887
16	1.4272	1.4272	1.4272	1.4272	0.8687

The initial values of the centers of the Gaussian antecedent MFs as well as of all $\bar{y}$ are given in Table 4.7. The initial values of the standard deviations of these Gaussian MFs were all set equal to the same value of $2\sigma_s = 0.5240$. The final values of all these parameters after six epochs of tuning are given in Tables 4.8 and 4.9.

The steepest descent (SD) RMSE, $\mathrm{RMSE}_s(\mathrm{SD})$, for each of the six epochs of tuning are:

$$\mathrm{RMSE}_s(\mathrm{SD}) = \{0.0548, 0.0431, 0.0322, 0.0261, 0.0237, 0.0232\} \tag{4.52}$$

These values were obtained by tuning for one epoch and then testing the design on the testing data, $s(1505)$, $s(1506), \ldots, s(2000)$; then, tuning for a second epoch and again testing on the testing data, and so forth, for six epochs. There is a substantial reduction in the RMSE from the first epoch to the sixth epoch, and one can see a leveling off of the RMSE at around the fifth epoch, so stopping the tuning after the sixth epoch is reasonable.

Recall that $\mathrm{RMSE}_s(\mathrm{OP}) = 0.0438$, whereas $\mathrm{RMSE}_s(\mathrm{SD}) = 0.0232$ for the final steepest descent design. It is rather amazing that a fuzzy system with only 16 rules vastly outperforms the 500-rule fuzzy system. This confirms the earlier comment that, although the one-pass method is simple it leads to a fuzzy system with too many rules (fuzzy basis functions).

Table 4.8 Final values for the centers of the Gaussian antecedent MFs and the centroid of the consequent set, after six epochs of tuning

Rule No.	Final value for centers of the four antecedent MFs				Final value for $\bar{y}^l$
1	0.4001	0.3613	0.3076	0.1694	0.4986
2	0.3075	0.2707	0.1988	1.5524	0.3860
3	0.4273	0.3821	1.3487	0.2121	0.0035
4	0.2586	0.3205	1.3205	1.3434	0.8631
5	0.3451	1.5229	0.3352	0.3297	0.6929
6	0.2942	1.5375	0.2316	1.4938	0.1211
7	0.3473	1.4700	1.4523	0.2704	0.4622
8	0.5727	1.1876	1.2624	1.3675	1.2695
9	1.5721	0.3604	0.3790	0.3960	0.9506
10	1.4782	0.2994	0.2817	1.4598	0.7021
11	1.4265	0.4093	1.3689	0.3367	0.2556
12	1.4560	0.2404	1.4518	1.4497	0.7644
13	1.4698	1.4427	0.3778	0.3500	0.6403
14	1.4748	1.4641	0.2593	1.4445	0.6486
15	1.4555	1.4210	1.3917	0.3730	0.3868
16	1.3964	1.4352	1.4933	1.6955	0.8715

Table 4.9 Final values for the standard deviations of the Gaussian antecedent MFs after six epochs of tuning

Rule No.	Final value for standard deviations			
1	0.5649	0.5254	0.4571	0.2268
2	0.4646	0.4094	0.2630	0.3728
3	0.5403	0.5075	0.7142	0.1399
4	0.3043	0.4044	0.6618	0.5931
5	0.5109	0.3561	0.4589	0.4267
6	0.4224	0.3640	0.2616	0.4736
7	0.4987	0.4718	0.5109	0.2772
8	0.6775	0.7240	0.6280	0.3527
9	0.2497	0.5214	0.5362	0.5489
10	0.4766	0.4106	0.3519	0.4872
11	0.4682	0.5493	0.6171	0.2966
12	0.5154	0.2404	0.5106	0.5082
13	0.4408	0.4884	0.5000	0.4111
14	0.4754	0.4926	0.2648	0.5216
15	0.4618	0.5299	0.5777	0.3553
16	0.5861	0.5408	0.4678	0.1198

4.3.4 A Change in the Measurements

Suppose the following situation occurs: two fuzzy system forecasters have just been designed (as in Sects. 4.3.2 and 4.3.3) based on perfect measurements, expecting that they will operate in the same noise-free environment; but, instead, the environment changes and they must now operate in a noisy measurement environment. How robust are the just-designed forecasters to this noisy measurement environment? To answer this question the two final fuzzy system forecasters were tested on noisy testing data, i.e., $x(k) = s(k) + n(k),\ k = 1505, 1506, \ldots, 2000$, where $n(k)$ is 0 dB uniformly distributed stationary noise (Fig. 4.4). This was done for a Monte Carlo set of 50 realizations. Figure 4.5 depicts the outputs of the two final singleton

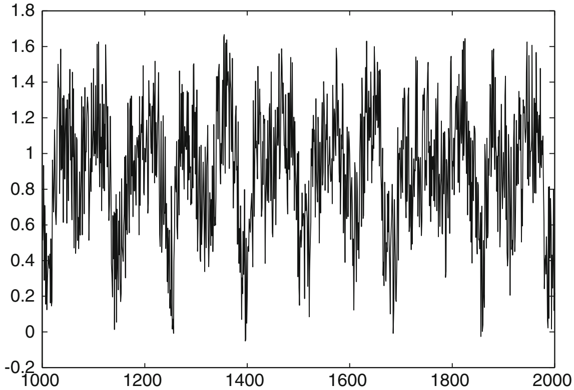


Fig. 4.4 One realization of 0 dB uniformly distributed noisy data, $x(1001), x(1002), \ldots, x(2000)$

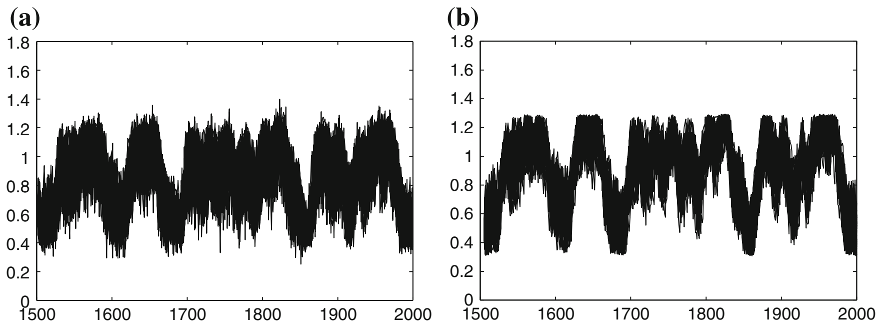


Fig. 4.5 Outputs of the: **a** steepest descent-designed 16-rule type-1 singleton fuzzy system forecaster, and **b** one-pass-designed 500-rule type-1 singleton fuzzy system forecaster. In both cases, there are 50 Monte Carlo realizations and the data from time-point 1504 to 2000 are corrupted by additive uniformly distributed noise with SNR = 0 dB

type-1 fuzzy system forecasters for the 50 Monte Carlo realizations (i.e., each part of Fig. 4.5 is the superimposition of 50 plots). Observe that the 500-rule one-pass-designed forecaster is more robust to the unexpected noise than is the 16-rule steepest descent-designed forecaster. Its standard deviation (i.e., the thickness of the superimposed plots) is visibly smaller than that of the steepest descent design. Having more rules makes the one-pass designed forecaster less susceptible to the additive measurement noise.

Note that the author is in no way advocating the use of a one-pass or a steepest descent-designed singleton fuzzy system forecaster in a noisy measurement environment. In fact, if one knows that only noisy measurements will be used in the fuzzy system forecaster, then such measurements should be accounted for and used during the design of the forecaster. This can be done in a number of different ways, one of which is explored next.

4.3.5 One-Pass Design: Non-singleton Fuzzification

Next, the one-pass design of Sect. 4.3.2 was repeated, but this time for a non-singleton fuzzy system forecaster that used the first 504 noisy data, $x(1001), x(1002), \ldots, x(1504)$ after which the remaining 496 data, $x(1505), x(1502), \ldots, x(2000)$, were used for testing the design.

The noise-free Mackey–Glass time series is still the one that is depicted in Fig. 4.3. As in Sect. 4.3.4, $x(k) = s(k) + n(k)$ $(k = 1001, 1002, \ldots, 2000)$ where $n(k)$ is uniformly distributed stationary additive noise, and SNR $= 0$ dB. The difference between the design in this section and the one in Sect. 4.3.2 is that here each fuzzy system forecaster is designed using the noisy training data, whereas in Sect. 4.3.2 each fuzzy system forecaster was designed using perfect training data.

As in Sect. 4.3.2, four antecedents were used to predict $x(k+1)$, namely, $x(k-3), x(k-2), x(k-1)$ and $x(k)$, and the performance of the design was again evaluated using a RMSE, namely,

$$\mathrm{RMSE}_{\mathrm{ns}} = \sqrt{\frac{1}{496} \sum_{k=1504}^{1999} \left[s(k+1) - y_{\mathrm{ns}}(\mathbf{x}^{(k)})\right]^2} \tag{4.53}$$

where $\mathbf{x}^{(k)} = [x(k-3), x(k-2), x(k-1), x(k)]^T$, $y_{\mathrm{ns}}(\mathbf{x})$ is obtained by substituting the bottom line of (4.2) into (4.1), and the subscripts "ns" on RMSE reminds us that this is a non-singleton design.

The Sect. 4.2.1.1 one-pass Data Assignment Method was used to construct 500 Zadeh rules from $s(1001), s(1002), \ldots, s(1504)$ The antecedent and consequent MFs were centered at the noisy measurements in each one of the 500 rules, and the standard deviation of the Gaussians was set equal to 0.1. Using the fact that signal-to-noise ratio $(\mathrm{SNR}) = 10\log_{10}(\sigma_s^2/\sigma_n^2)$ (from which it follows that $\sigma_n = \sigma_s/10^{\mathrm{SNR}/20}$), σ_X was set[11] equal to σ_n. Once σ_X was set, this completely established the Mamdani non-singleton fuzzy system $y_{\mathrm{ns}}(\mathbf{x})$. Then, $x(1505), x(1506), \ldots, x(2000)$ were used to compute the one-pass $\mathrm{RMSE_{ns}}$ in (4.53). This entire process was repeated 50 times using 50 independent sets of 1000 data points, at the end of which 50 $\mathrm{RMSE_{ns}}(\mathrm{OP})$ values were available. The average value and standard deviation for these 50 $\mathrm{RMSE_{ns}}$ values are $\overline{\mathrm{RMSE}}_{\mathrm{ns}}(\mathrm{OP}) = 0.1304$ and $\sigma_{\mathrm{RMSE_{ns}(OP)}} = 0.0064$.

How does this one-pass-designed non-singleton fuzzy system forecaster compare with a one-pass-designed singleton fuzzy system forecaster that also used $x(1001), x(1002), \ldots,$ and $x(1504)$? The 500 rules for both designs are exactly the same for each of the 50 realizations of the data (because rules were extracted from the noisy data); however, the non-singleton fuzzy system forecaster includes the effect of the noisy measurements by means of its pre-filter (Fig. 3.7), whereas the singleton fuzzy system forecaster does not. The RMSE_s for each of the one-pass-designed singleton fuzzy system forecasters was computed using

$$\mathrm{RMSE}_s = \sqrt{\frac{1}{496}\sum_{k=1504}^{1999}\left[s(k+1) - y_s(\mathbf{x}^{(k)})\right]^2} \tag{4.54}$$

The average value and standard deviation for these 50 $\mathrm{RMSE}_s(\mathrm{OP})$ values are $\overline{\mathrm{RMSE}}_s(\mathrm{OP}) = 0.1882$ and $\sigma_{\mathrm{RMSE}_s(\mathrm{OP})} = 0.0079$. Observe that $\overline{\mathrm{RMSE}}_s(\mathrm{OP})$ is more than 40% larger than $\overline{\mathrm{RMSE}}_{\mathrm{ns}}(\mathrm{OP})$. So, accounting for the noise by using non-singleton fuzzification can result in a substantial improvement in RMSE performance, even for a one-pass design.

Figure 4.6 depicts this more vividly. It depicts the *output* of the two one-pass-designed type-1 fuzzy system forecasters for 50 Monte Carlo realizations (i.e., each part of Fig. 4.6 is the superimposition of 50 plots). Clearly, the 500-rule one-pass-designed non-singleton type-1 fuzzy system forecaster has a smaller standard deviation (i.e., the thickness of the superimposed plots) than the 500-rule one-pass-designed singleton type-1 fuzzy system forecaster. The pre-filtering by the non-singleton type-1 fuzzy system forecaster makes a substantial difference.

[11] Note that, because this was a simulation, the noise-free signal, $s(k)$ was accessible; hence, it was possible to compute σ_s, and since *SNR* was also assumed known, σ_X could be computed. In practice, however, $s(k)$ would not be accessible, and *SNR* would be unknown. In the steepest-descent method, described in Sect. 4.3.6, σ_X is estimated directly from the data. The more realistic case, when σ_s and *SNR* are unknown is covered in Sect. 10.3.2.

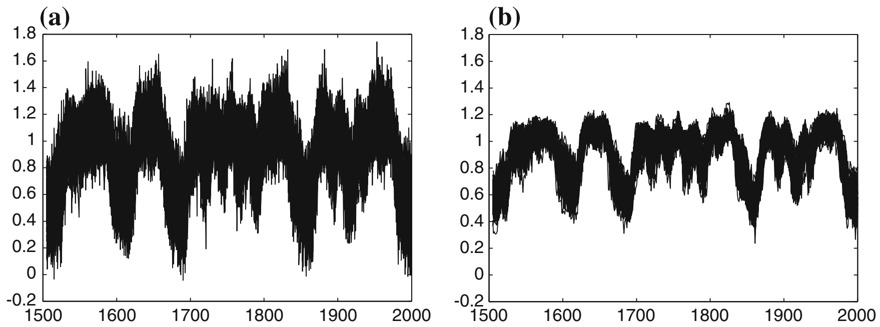


Fig. 4.6 Outputs of the: **a** one-pass-designed singleton fuzzy system forecaster, and **b** one-pass-designed non-singleton fuzzy system forecaster. In both cases, there are 50 Monte Carlo realizations and the data from time point 1504–2000 are corrupted by additive uniformly distributed stationary noise with SNR = 0 dB

4.3.6 Derivative-Based Design: Non-singleton Fuzzification

Next, a non-singleton type-1 fuzzy system is compared with a singleton type-1 fuzzy system when both are designed using steepest descent. Because the non-singleton fuzzy system shares most of the same parameters as the singleton fuzzy system, the Sect. 4.1 partially dependent design approach was used to design the non-singleton fuzzy system.

As in the Sect. 4.3.3 steepest descent design, only two fuzzy sets were used for each of the four antecedents, so that there are again only 16 rules. Each rule is characterized by eight antecedent MF parameters (the mean and standard deviation for each of the four Gaussian MFs) and one consequent parameter, $\bar{y}$. The initial location of each Gaussian antecedent MF was based on the mean, m_x, and the standard deviation, σ_x, of the data in the 504 noisy training samples, $x(1001), x(1002), \ldots, x(1504)$. More specifically, the initial means of each and every antecedent's two Gaussian MFs were chosen as $m_x - 2\sigma_x$ or $m_x + 2\sigma_x$, respectively, and their initial standard deviations were chosen as $2\sigma_x$. The center of each consequent's MF, $\bar{y}^i$ $(i = 1, \ldots, 16)$, was again initially chosen to be a random number from the interval [0, 1].

For the non-singleton type-1 fuzzy system, each of the four noisy input measurements was modeled using a Gaussian MF. Two choices were possible: (1) use a different standard deviation for each of the four input measurement MFs, or (2) use the same standard deviation for each of the four input measurement MFs. Both approaches were tried and very similar results were obtained for them. Because the additive noise is stationary, using the same standard deviation for each of the four input measurement MFs should suffice; hence, results are only presented for that choice.

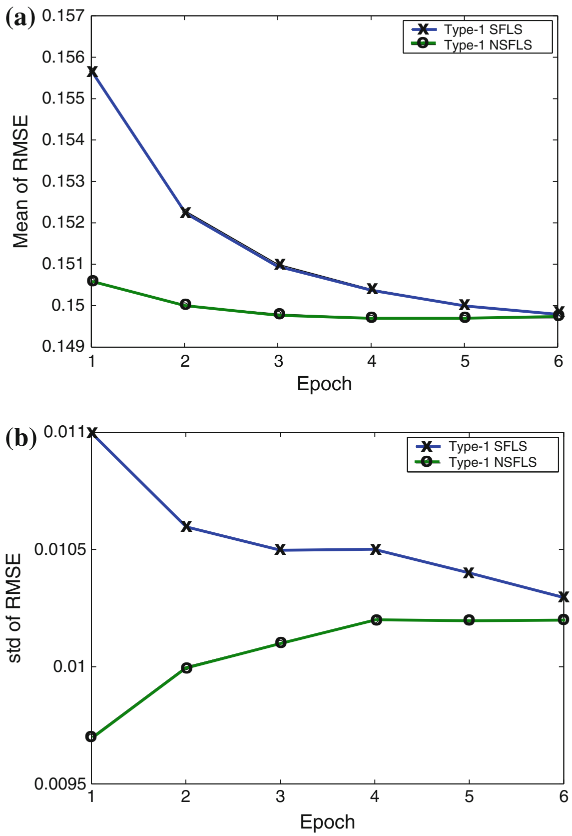


Fig. 4.7 The mean and standard deviations of $\mathrm{RMSE}_s(\mathrm{SD})$ and $\mathrm{RMSE}_{\mathrm{ns}}(\mathrm{SD})$, averaged over 50 Monte Carlo designs. Tuning was performed in each realization for six epochs. **a** mean values, and **b** standard deviation values. Note that "FLS" is equivalent to "fuzzy system."

Each fuzzy system was tuned using a steepest descent algorithm in which all of the learning parameters were set equal to the same $\beta_0 = 0.2$. Training and testing were carried out for six epochs. After each epoch the testing data were used to evaluate how each fuzzy system performed, by computing $\mathrm{RMSE}_s(\mathrm{SD})$ using (4.54) and $\mathrm{RMSE}_{\mathrm{ns}}(\mathrm{SD})$ using (4.53). This entire process was repeated 50 times using 50 independent sets of 1000 data points,[12] at the end of which 50 $\mathrm{RMSE}_s(\mathrm{SD})$ and 50 $\mathrm{RMSE}_{\mathrm{ns}}(\mathrm{SD})$ values were obtained.

The mean and standard deviations of $\mathrm{RMSE}_s(\mathrm{SD})$ and $\mathrm{RMSE}_{\mathrm{ns}}(\mathrm{SD})$ are plotted in Fig. 4.7 for each of the six epochs. Observe that the type-1 non-singleton fuzzy system always outperforms the type-1 singleton fuzzy system, *although not by much*, as a careful examination of the scales on the two plots reveals. In fact, after six epochs of training the performances of the two designs are just about the same; i.e., the final $\overline{\mathrm{RMSE}}_s(\mathrm{SD}) \approx$ the final $\overline{\mathrm{RMSE}}_{\mathrm{ns}}(\mathrm{SD}) \approx 0.15$. Note that the minimum value of the

[12]Tables like Tables 4.7, 4.8 and 4.9 are not included here, or in later chapters, because when the training data are noisy 50 Monte-Carlo designs are performed, and three tables would need to be shown for each of the 50 designs.

average RMSE is just about reached after the first epoch by the non-singleton fuzzy system, whereas it takes five epochs for the singleton fuzzy system to reach this value. This suggests that the non-singleton type-1 fuzzy system could be used in a real-time adaptive environment, whereas the singleton type-1 fuzzy system should not be.

It is possible to explain why the steepest descent-designed singleton and non-singleton fuzzy systems achieved the same final performance. At the end of Example 3.7, it was explained that a non-singleton fuzzy system that uses Gaussian MFs and product t-norm is equivalent to a singleton fuzzy system whose MFs are given in (3.31). Observe, from (3.31), that σ_X^2 and $\sigma_{F_k^l}^2$ always occur as the sum $\sigma_{F_k^l}^2 + \sigma_X^2$; hence, tuning them individually (as has been done) cannot be expected to give better results than tuning their sum, as is essentially done in the type-1 design. Consequently, in this case, the tuned singleton and non-singleton fuzzy systems should lead to the same results.

In a one-pass design, on the other hand, there is no tuning. Fixing σ_X^2 in the non-singleton one-pass design can cause that design to outperform the singleton one-pass design because in a one-pass design σ_X^2 and $\sigma_{F_k^l}^2$ are set separately.

It is conjectured that better performance can be obtained for the non-singleton forecaster by using the totally independent design approach, in which all of the parameters of the non-singleton forecaster are tuned.

4.3.7 Final Remark

Another reason for the not-so-spectacular improvement in performance by the type-1 non-singleton fuzzy system is that all of the uncertainties have not been properly accounted for. The training data are noisy, but there is no way to properly account for this in the antecedent MFs of a type-1 fuzzy system. The best that can be done using a type-1 non-singleton fuzzy system is to account for the noise in the measurements that excite the fully designed non-singleton fuzzy system through the filtering action of non-singleton fuzzification; but, that is just not good enough! This case study is returned to in Chap. 10, where for the first time, noisy training data will he handled correctly.

4.4 Case Study: Knowledge Mining Using Surveys

Knowledge mining,[13] as used in this book, means extracting information in the form of IF–THEN rules from people. These rules can be modeled using a fuzzy system, which can then be used as a[14] fuzzy logic advisor (FLA) to assist in making

[13]Another term for this is *knowledge engineering*.

[14]In this section the term "fuzzy logic advisor" is retained instead of calling it a "fuzzy system advisor", because all earlier publications on this subject use "fuzzy logic advisor".

decisions or judgments. By "judgment" is meant an assessment of the *level* of a variable of interest. For example, in everyday social interaction, each of us is called upon to make judgments about the meaning of another's behavior (e.g., kindness, generosity, flirtation, harassment). Such judgments are far from trivial, since they often affect the nature and direction of the subsequent social interaction and communications. Although a variety of factors may enter into our decision, behavior (e.g., touching, eye contact) is apt to play a critical role in assessing the level of the variable of interest.

The techniques introduced in this section should be applicable to many situations in which rule-based decision-making is needed and inputs and outputs are words (judgments), e.g., environmental engineering, social science, security, etc.

4.4.1 Methodology for Knowledge Mining

In developing a FLA, it is useful to adopt the following methodology [adapted from Mendel et al. (1999, pp. 468–472)]:

1. *Identify the behavior of interest*. This step, although obvious, is highly application dependent. For social judgments the behaviors of kindness, generosity, flirtation, and harassment have already been mentioned; other social variables of interest might be level of violence or amount of sexually explicit material in a television program. For engineering judgments, variables of interest might include toxicity, video quality, sound quality, environmental contamination level, global warming, etc.
2. *Determine the indicators of the behavior of interest*. This sometimes requires:
   (a) Establishing a list of candidate indicators [e.g., for flirtation, as in Mendel, et al. (1999), six candidate indicators are touching, eye contact, acting witty, primping, smiling, and complementing].
   (b) Conducting a survey in which a representative population is asked to rank-order in importance the indicators on the list of candidate indicators. In some applications it may already be known what the relative importance of the indicators is, in which case a survey is not necessary.
   (c) Choosing a meaningful subset of the indicators, because not all of them may be important. In Step 6, where people are asked to provide consequents for a collection of IF–THEN rules by means of a survey, the survey must be kept manageable, because most people do not like to answer lots of questions; hence, it is very important to focus on the truly significant indicators. Factor analysis, from statistics, can be used to help establish the relative significance of indicators.
3. *Establish scales for each indicator and the behavior of interest*. If an indicator is a physically measurable quantity (e.g., temperature, pressure), then the scale is associated with the expected range between the minimum and maximum values

for that quantity. On the other hand, many indicators are not measurable by means of instrumentation (e.g., touching, flirtation, harassment, video quality, etc.). Such indicators need to have a scale associated with them, or else it will not be possible to design or activate a FLA. Commonly used scales are 1 through 5, 0 through 5, 0 through 10, etc.

4. *Establish names and interval information for each of the indicator's fuzzy sets and behavior of interest's fuzzy sets*. The issues here are:

   (a) What vocabulary should be used and what should its size be so that the MFs for the vocabulary cover the scale (e.g., 0–10) and provide the user of the FLA with a user-friendly interface?
   (b) What is the smallest number of fuzzy sets that should be used for each indicator and behavior of interest for establishing rules?
   (c) How should word data be collected so that methodological uncertainties are not introduced?

   Surveys can be used to provide answers to the first two questions. Section 5.2 provides an answer to the third question.

5. *Establish the rules*. Rules are the heart of the FLA; they link the indicators of a behavior of interest to that behavior. The following issues need to be addressed:

   (a) How many antecedents will the rules have? As mentioned earlier, people generally do not like to answer complicated questions; so, using rules that have either one or two antecedents is advocated. An interesting (non-engineering) interpretation for a two-antecedent rule is that it provides the *correlation* effect that exists in the mind of the survey respondent between the two antecedents. Psychologists have told the author that it is just about impossible for humans to correlate more than two antecedents (indicators) at a time, and that even correlating two antecedents at a time is difficult. Using only one or two antecedents does not mean that a person does not use more than this number of indicators to make a judgment; it means that a person uses the indicators one or two at a time (this should be viewed as a *conjecture*). This suggests that the overall architecture for the FLA should be parallel or hierarchical.
   (b) How many rule bases need to be established? Each rule base leads to its own FLA. When there is more than one rule base, each of the advisors is a FL *sub-advisor*, and the outputs of these sub-advisors can be combined to create the structure of the overall FLA. If, e.g., it has been established that four indicators are equally important for the judgment of flirtation, then there could be up to four single-antecedent rule bases as well as six two-antecedent rule bases. A decision must be made about which of the rule bases would actually be used. This can be done by means of another survey in which the respondents are asked to rank-order the rule bases in order of importance. Later, when (and if) the outputs of the different rule bases are combined, they can be weighted using the results of this step.

There is a very important reason for using sub-advisors for a FLA (Mendel and Wu 2010). It is very unlikely that all of the important indicators for a social judgment will occur at the same time. For example, for flirtation, if touching, eye contact, acting witty and primping have been established as its four most important indicators, it is very unlikely that in a new flirtation situation all four will occur simultaneously. From your own experiences in flirting, can you recall a situation when someone was touching you, made eye contact with you, was acting witty, and was also primping, all at the same time? Not very likely! Note that a missing observation is not the same as an observation of zero value; hence, even if it were possible to create four antecedent rules, none of those rules could be activated if one or more of the indicators had a missing observation. It is, therefore, very important to have subadvisors that will be activated when only one or two of these indicators are occurring.

6. *Survey people (experts) to provide consequents for the rules*. If, e.g., the antecedent in a single-antecedent rule has five fuzzy sets associated with it, then respondents would be asked five questions. For two-antecedent rules, where each antecedent is again described by five fuzzy sets, there would be 25 questions. The order of the questions should be randomized so that respondents do not correlate their answers from one question to the next. Each single-antecedent rule is associated with a question of the form:

   IF the antecedent is (state the name of one of the antecedent's fuzzy sets), THEN there is (state the name of one of the consequent's fuzzy sets) of the behavior.

   Each two-antecedent rule is associated with a question of the form:

   IF antecedent 1 is (state the name of one of antecedent 1's fuzzy sets) and antecedent 2 is (state the name of one of antecedent 2's fuzzy sets), THEN there is (state the name of one of the consequent's fuzzy sets) of the behavior.

   The respondent is asked to choose one of the given names for the consequent's fuzzy sets (established in Step 4). These rule base surveys will lead to rule consequent histograms, because everyone will not answer a question the same way.

4.4.2 Survey Results

Survey results are presented here that are used below as well as in Chap. 10 to design FLAs. This is done for a generic behavior to illustrate the design procedures, so as not to get lost in the details of a specific social or engineering behavior.

The following five terms are used for antecedents and consequent: *none to very little*, *some*, *a moderate amount*, *a large amount*, and *a maximum amount*. Table 4.10 summarizes the data collected from the Step 4 survey for these labels.

Table 4.10 Processed survey results for labels of fuzzy sets

No.	Range label	Mean		Standard deviation	
		Start (a)	End (b)	Start (a)	End (b)
		m_a	m_b	σ_a	σ_b
1	*None to very little* (NVL)	0	1.99	0	0.81
2	*Some* (S)	2.54	5.25	0.91	1.37
3	*A moderate amount* (MOA)	3.64	6.46	0.88	0.86
4	*A large amount* (LA)	6.48	8.75	0.75	0.60
5	*A maximum amount* (MAA)	8.55	10	0.75	0

47 students provided interval $[a, b]$ end-points a and b (on a scale of 0–10) for each of the five words. Exactly how these data were obtained is explained in Sect. 5.2 The mean and standard deviation of the two end-points for the 47 data values are summarized in Table 4.10.

Our FLA is limited to rule bases for one- and two-antecedent rules. In the spirit of generic results, x_1 and x_2 are used to denote the generic antecedents and y is used to denote the generic consequent for these rules. Tables 4.11, 4.12 and 4.13 provide the data collected from the 47 respondents to the Step 6 surveys. The antecedents for each rule appear in the parentheses after the rule number.

4.4.3 Determining Type-1 Fuzzy Sets from Survey Results

In order to design and implement a FLA, the next step is to associate type-1 MFs with the Table 4.10 interval data. Figure 4.8 depicts the MFs[15] chosen by us for the five linguistic labels (*none to very little*, *some*, *a moderate amount*, *a large amount*, and *a maximum amount*). Triangular MFs have been chosen for the three interior labels (*some*, *a moderate amount*, and *a large amount*) and piecewise-linear MFs have been chosen for the leftmost and rightmost labels (*none to very little* and *a maximum amount*). These choices were made because it is relatively easy, as is explained next, to go from the data in Table 4.10 to such MFs.

The triangular MFs were constructed as follows (values for all group statistics are given in Table 4.10): their (1) apex was located at $(m_a + m_b)/2$; (2) left-end base point was located at $m_a - \sigma_a$, and (3) right-end base point was located at $m_b + \sigma_b$.

The piecewise-linear MF for *none to very little* was constructed as follows: (1) since there is no uncertainty about its left-hand end-point, it begins at zero with

[15]By using the group statistics for the end-points of a word, these MFs only model the inter-uncertainties of a word (the uncertainties that the group of subjects has about the word). Two kinds of word uncertainties are explained in Chap. 5. Interval type-2 fuzzy sets are able to model both the word's inter-uncertainties and the intra-uncertainties (the uncertainties that each subject has about the word); one way to do this is discussed in Sect. 10.4.1.

Table 4.11 Histogram of survey responses for single-antecedent rules between indicator x_1 and consequent y

Rule No.	Consequent					c^l_{avg}
	None to very little (NVL)	*Some* (S)	*A moderate amount* (MOA)	*A large amount* (LA)	*A maximum amount* (MAA)	
1 (NVL)	42	3	2	0	0	1.53
2 (S)	33	12	0	2	0	2.19
3 (MOA)	12	16	15	3	1	3.97
4 (LA)	3	6	11	25	2	6.18
5 (MAA)	3	6	8	22	8	6.54

Entries denote the number of respondents out of 47 that chose the consequent. c^l_{avg} is the weighted average of the responses, given by (4.56)

Table 4.12 Histogram of survey responses for single-antecedent rules between indicator x_2 and consequent y

Rule No.	Consequent					c^l_{avg}
	None to very little (NVL)	*Some* (S)	*A moderate amount* (MOA)	*A large amount* (LA)	*A maximum amount* (MAA)	
1 (NVL)	36	7	4	0	0	1.94
2 (S)	26	17	4	0	0	2.55
3 (MOA)	2	16	27	2	0	4.65
4 (LA)	1	3	11	22	10	6.94
5 (MAA)	0	3	7	17	20	7.62

Entries denote the number of respondents out of 47 that chose the consequent. c^l_{avg} is the weighted average of the responses, given by (4.56)

an amplitude of unity; (2) the unity amplitude continues to the right until it reaches $m_b = 1.99$; (3) the point $m_b + \sigma_b$ is located on the horizontal axis; and (4) the two points established in Steps (2) and (3) are connected by a straight line.

Finally, the piecewise-linear MF for *a maximum amount* was constructed as follows: (1) since there is no uncertainty about its right-hand end-point, it ends at 10 with an amplitude of unity; (2) the unity amplitude continues to the left until it reaches $m_a = 8.55$; (3) the point $m_a - \sigma_a$ is located on the horizontal axis; and, (4) the two points established in Steps (2) and (3) are connected by a straight line.

There is uncertainty associated with the use of $m_a - \sigma_a$ and $m_b + \sigma_b$; e.g. why not use $m_a - 0.5\sigma_a$ or $m_a - 2\sigma_a$, instead of $m_a - \sigma_a$, and $m_b + 0.5\sigma_b$ or $m_b + 1.5\sigma_b$ instead of $m_b + \sigma_b$? This uncertainty cannot be captured using type-1 fuzzy sets; however, as is explained in Sect. 10.4.1, it can be captured using type-2 fuzzy sets.

Table 4.13 Histogram of survey responses for two-antecedent rules between indicators x_1 and x_2 and consequent y

Rule No.	Consequent					c_{avg}^l
	None to very little (NVL)	*Some* (S)	*A moderate amount* (MOA)	*A large amount* (LA)	*A maximum amount* (MAA)	
1 (NVL/NVL)	38	7	2	0	0	1.77
2 (NVL/S)	33	11	3	0	0	2.10
3 (NVL/MOA)	6	21	16	4	0	4.32
4 (NVL/LA)	0	12	26	8	1	5.30
5 (NVL/MAA)	0	9	16	19	3	6.13
6 (S/NVL)	31	11	4	1	0	2.32
7 (S/S)	17	23	7	0	0	3.16
8 (S/MOA)	0	19	19	8	1	5.16
9 (S/LA)	1	8	23	13	2	5.66
10 (S/MAA)	0	7	17	21	2	6.19
11 (MOA/NVL)	7	23	16	1	0	4.04
12 (MOA/S)	5	22	20	0	0	4.17
13 (MOA/MOA)	2	7	22	15	1	5.62
14 (MOA/LA)	1	4	13	17	12	6.82
15 (MOA/MAA)	0	4	12	24	7	6.85
16 (LA/NVL)	7	13	21	6	0	4.51
17 (LA/S)	3	11	23	10	0	5.10
18 (LA/MOA)	0	3	18	18	8	6.64
19 (LA/LA)	0	1	9	17	20	7.66
20 (LA/MAA)	1	2	6	11	27	7.84
21 (MAA/NVL)	2	16	18	11	0	5.13
22 (MAA/S)	2	9	22	13	1	5.47
23 (MAA/MOA)	0	3	15	18	11	6.90
24 (MAA/LA)	0	1	7	17	22	7.83
25 (MAA/MAA)	0	2	3	12	30	8.24

Entries denote the number of respondents out of 47 that chose the consequent. c_{avg}^l is the weighted average of the responses, given by (4.56)

The centroids of the five fuzzy sets (computed for the MFs that are depicted in Fig. 4.8, to two significant figures), which are used below in (4.56), are:

$$\begin{aligned} &c_1 = c_{\text{NVL}} = 1.18,\ c_2 = c_{\text{S}} = 4.05,\ c_3 = c_{\text{MOA}} = 5.04, \\ &c_4 = c_{\text{LA}} = 7.57,\ c_5 = c_{\text{MAA}} = 9.10 \end{aligned} \tag{4.55}$$

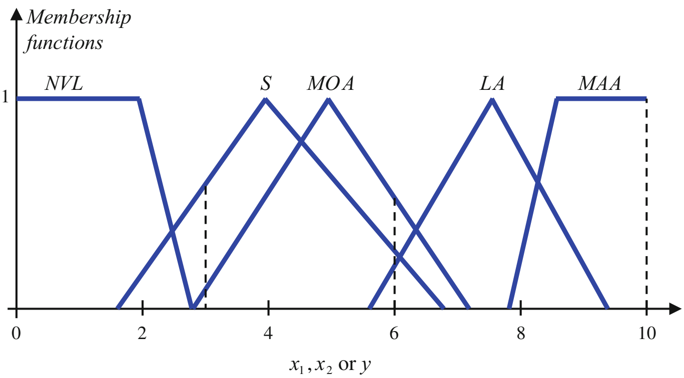


Fig. 4.8 MFs for the five linguistic labels *none to very little* (*NVL*), *some* (*S*), *a moderate amount* (*MOA*), *a large amount* (*LA*), and *a maximum amount* (*MAA*). The *dashed vertical lines* at 3 and 6 are used in Example 4.9

4.4.4 What Does One Do with a Histogram of Responses?

Tables 4.11, 4.12 and 4.13 reveal that each rule has a histogram of responses,[16] which raises the question "what should be done with this information?" Three possibilities come to mind:

1. Keep the response chosen by the largest number of experts. So, for example, the consequent used for Rule 4 in Table 4.11 would be *a large amount*.
2. Find a weighted average of the rule consequents for each rule. In this case, the consequent of each rule is treated as a number, c_{avg}^{l} (e.g., a two-antecedent rule is now interpreted as IF x_1 is F^i and x_2 is F^j, THEN $y = c_{\text{avg}}^{l}$), where

$$c_{\text{avg}}^{l} = \frac{\sum_{i=1}^{5} c_i w_i^l}{\sum_{i=1}^{5} w_i^l} \tag{4.56}$$

   In (4.56), c_i is the centroid of the ith consequent set and w_i^l is the weight associated with the ith consequent for the lth rule. The count entries in Tables 4.11, 4.12 and 4.13 are used as the weights.
3. Preserve the distributions of the expert responses for each rule.

---

[16]As is mentioned in Mendel and Wu (2010, Sect. 8.2.3), inevitably there are bad responses and outliers in survey histograms that need to be removed before the histograms are used. That reference explains three histogram data preprocessing steps: (1) *bad data processing*, which removes gaps (a zero between two non-zero values) in a group of subject's responses; (2) *outlier processing*, which removes points that are unusually too large or too small; and (3) *tolerance limit processing*, which only keeps data in a tolerance interval. Refer to Mendel and Wu (2010, Sect. 8.2.3) for the details. To keep things as simple as possible in this book, histogram data preprocessing is not used.

Obviously, preserving the distributions of the expert responses for each rule makes maximum use of the uncertainties associated with rule consequents; however, it is computationally very costly. This strategy is discussed later in Sect. 4.4.6.

Keeping the response chosen by the largest number of experts completely ignores the uncertainties associated with rule consequents; hence, this strategy is not adopted.

Finally, finding a weighted average of the rule consequents for each rule uses some aspects of the uncertainties that are associated with rule consequents, does not lead to excessive computations, and is the strategy that is adopted and discussed further in Sect. 4.4.5.

4.4.5 Averaging the Responses: Consensus FLAs

To begin c_{avg}^l must be computed for the survey results in Tables 4.11, 4.12 and 4.13. These results are given in the right-most column of those tables. Note that c_{avg}^l plays the role of λ^l in the FBF expansion (4.1), where $\phi_l(\mathbf{x})$ are given in the top line of (4.2). From Tables 4.11, 4.12 and 4.13, and Eqs. (4.1) and (4.2), it is straightforward to compute the three consensus singleton Mamdani FLAs $y_{c1}(x_1)$,$y_{c1}(x_2)$ and $y_{c1}(x_1, x_2)$ (where the subscript "$c1$" denotes "consensus type-1"), as:

$$y_{c1}(x_1) = \sum_{l=1}^{5} c_{\text{avg}}^l \phi_l(x_1) = \sum_{l=1}^{5} c_{\text{avg}}^l \left[\frac{\mu_{F_1^l}(x_1)}{\sum_{l=1}^{5} \mu_{F_1^l}(x_1)}\right], \tag{4.57}$$

c_{avg}^l from Table 4.11

$$y_{c1}(x_2) = \sum_{l=1}^{5} c_{\text{avg}}^l \phi_l(x_2) = \sum_{l=1}^{5} c_{\text{avg}}^l \left[\frac{\mu_{F_1^l}(x_2)}{\sum_{l=1}^{5} \mu_{F_1^l}(x_2)}\right], \tag{4.58}$$

c_{avg}^l from Table 4.12

$$\begin{aligned} y_{c1}(x_1, x_2) &= \sum_{l=1}^{25} c_{\text{avg}}^l \phi_l(x_1, x_2) \\ &= \sum_{l=1}^{25} c_{\text{avg}}^l \left[\frac{\mu_{F_1^l}(x_1) \times \mu_{F_2^l}(x_2)}{\sum_{l=1}^{25} \mu_{F_1^l}(x_1) \times \mu_{F_2^l}(x_2)}\right], \end{aligned} \tag{4.59}$$

c_{avg}^l from Table 4.13

The MFs used in these equations for $\mu_{F_1^l}(x_1)$ and $\mu_{F_2^l}(x_2)$ are the ones in Fig. 4.8. Plots of $y_{c1}(x_1), y_{c1}(x_2)$, and $y_{c1}(x_1, x_2)$ are depicted in Figs. 4.9, 4.10 and 4.11, respectively.

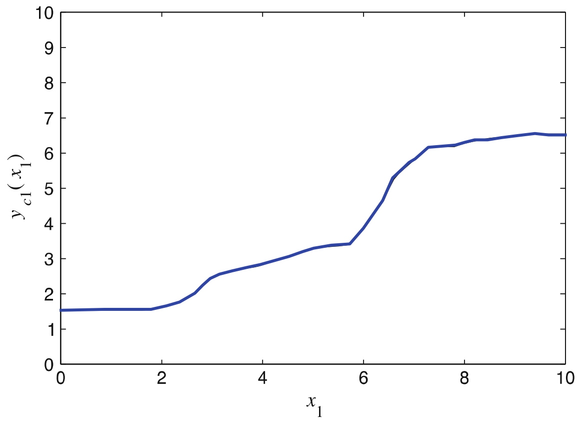


Fig. 4.9 FLA $y_{c1}(x_1)$

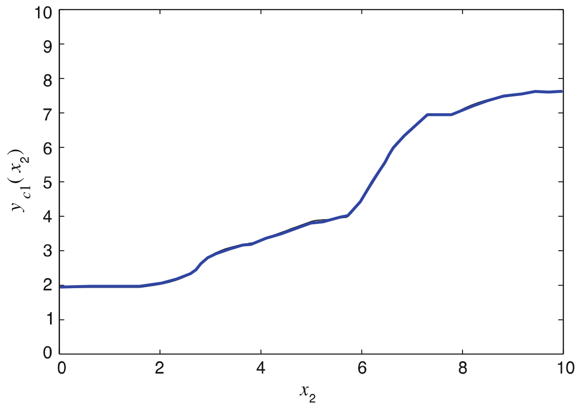


Fig. 4.10 FLA $y_{c1}(x_2)$

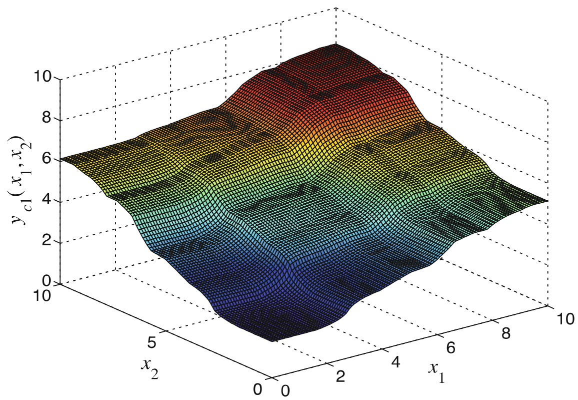


Fig. 4.11 FLA $y_{c1}(x_1, x_2)$

Example 4.9 A sample calculation of $y_{c1}(x_1, x_2)$ when $(x_1, x_2) = (3, 6)$ is provided in this example. Observe, from Fig. 4.8, that when $x_1 = 3$ two subsets are fired, S and MOA; their firing degrees are 0.60 and 0.11, respectively. When $x_2 = 6$ three subsets are fired, S, MOA, and LA; their firing degrees are 0.23, 0.58, and 0.14, respectively. It follows, therefore, that the six rules whose antecedent pairs are (S, S), (S, MOA), (S, LA), (MOA, S), (MOA, MOA), and (MOA, LA) are the ones fired. From Table 4.13, observe that these are Rules 7, 8, 9, 12, 13, and 14. The firing level for each of these rules is obtained by multiplying the rule's respective antecedent firing degrees, e.g., the firing degree for R_Z^8 is $0.60 \times 0.58 = 0.35$. Consequently, $y_{c1}(3, 6)$ is computed using (4.59), as:

$$\begin{aligned} y_{c1}(3,6) &= \left[\frac{1}{(0.14+0.35+0.09+0.02+0.06+0.01)}\right] \\ &\quad \times [0.14 \times 3.16 + 0.35 \times 5.16 + 0.09 \times 5.66 \\ &\quad + 0.02 \times 4.17 + 0.06 \times 5.62 + 0.01 \times 6.82] \\ &= 3.25/0.67 \\ &= 4.85 \end{aligned} \tag{4.60}$$

Note that single-antecedent FLAs for x_1 and x_2 implicitly assume independence between x_1 and x_2, whereas a zero value for either x_1 or x_2 in the two-antecedent FLA is not equivalent to independence between x_1 and x_2; hence, $y_{c1}(x_1, 0) \neq y_{c1}(x_1)$ and $y_{c1}(0, x_2) \neq y_{c1}(x_2)$.

4.4.6 Preserving All of the Responses

Each possible response to every question can be considered to form one rule of a FLA, and since one rule in a FLA can have only one consequent, different responses to the same question can be considered as rules from different FLAs. All the expert responses taken together can be viewed as a collection of many different type-1 FLAs, each corresponding to one combination of expert responses.

For each of the single-antecedent results in Tables 4.11 and 4.12, there are $5^5 = 3125$ possible type-1 FLAs. For the two-antecedent results in Table 4.13, there are $5^{25} = 9{,}765{,}625$ possible type-1 FLAs. Each possible consequent for every rule can be assigned a weight that is equal to the percentage of experts who voted in favor of it; hence, each rule in every one of the 3125 or 9,765,625 FLAs can be assigned the weight of its consequent, and each FLA can be assigned a weight equal to the t-norm of the weight of its 5 or 25 rules.

Consequently, one survey can be represented as a collection of many different type-1 FLAs, each having this weight associated with it. While this may be somewhat practical for the single-antecedent survey results, it is impractical for the two-antecedent survey results. For additional discussions about this approach, see Karnik and Mendel (1998). A practical alternative to using the entire collection of type-1 FLAs is explained in Sect. 10.4; it uses interval type-2 fuzzy sets.

4.4.7 On Multiple Indicators

For the sake of this discussion (Mendel and Wu 2010, Sect. 8.3.4) assume that the judgment is a social judgment (SJ) described by four indicators I_1, I_2, I_3 and I_4, and that the following ten FLAs have been created:

FLA_1: IF I_1 is ____________, THEN SJ is ____________.
FLA_2: IF I_2 is ____________, THEN SJ is ____________.
FLA_3: IF I_3 is ____________, THEN SJ is ____________.
FLA_4: IF I_4 is ____________, THEN SJ is ____________.
FLA_5: IF I_1 is ____________ and I_2 is ____________, THEN SJ is ____________.
FLA_6: IF I_1 is ____________ and I_3 is ____________, THEN SJ is ____________.
FLA_7: IF I_1 is ____________ and I_4 is ____________, THEN SJ is ____________.
FLA_8: IF I_2 is ____________ and I_3 is ____________, THEN SJ is ____________.
FLA_9: IF I_2 is ____________ and I_4 is ____________, THEN SJ is ____________.
FLA_{10}: IF I_3 is ____________ and I_4 is ____________, THEN SJ is ____________.

These ten FLAs can be used as follows:

1. When only one indicator is observed, only one single-antecedent FLA from FLA_1 to FLA_4 is activated.
2. When only two indicators are observed, only one two-antecedent FLA from FLA_5 to FLA_{10} is activated.
3. When more than two indicators are observed, the output is computed by aggregating the outputs of the activated two-antecedent FLAs. For example, when the observed indicators are I_1, I_2 and I_4, three two-antecedent FLAs—FLA_5, FLA_7 and FLA_9—are activated, and each one gives an SJ level. The final output is some kind of aggregation of the results from these three FLAs. An intuitive approach is to survey the subjects about the relative importance of the four indicators and then determine the relative importance of FLA_5–FLA_{10}. These relative importance values can then be used as the weights for FLA_5–FLA_{10}, and the SJ level can then be computed by a weighted average.

A diagram of the proposed FLA architecture for different numbers of indicators is shown in Fig. 4.12.

4.4.8 How to Use a FLA

Each FLA that has been designed above can be referred to as a *consensus* FLA, because it is obtained by using survey results from a population of people. This section describes how one can use the resulting FLAs.

Figure 4.13 depicts one way to use a FLA to advise an individual about a social judgment. It assumes that an individual is given the same questionnaire that was used in Step 6 of the knowledge mining process, which led to the consensus FLA.

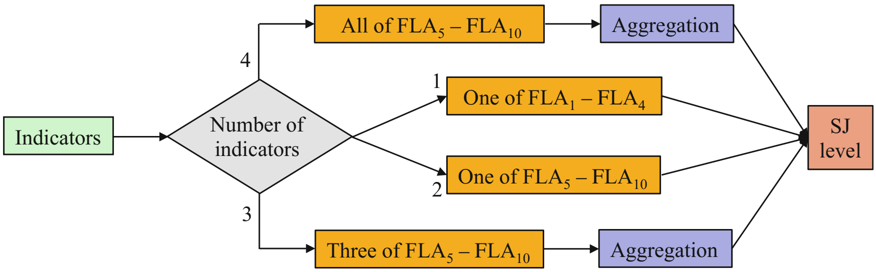


Fig. 4.12 A FLA architecture for one to four indicators [Mendel and Wu (2010, Fig. 8.10), © IEEE 2010]

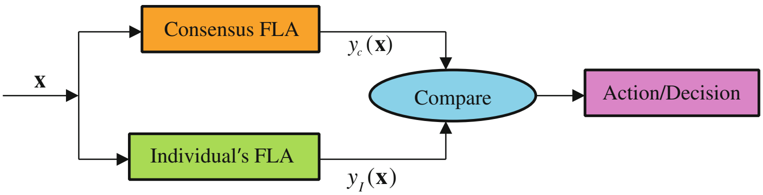
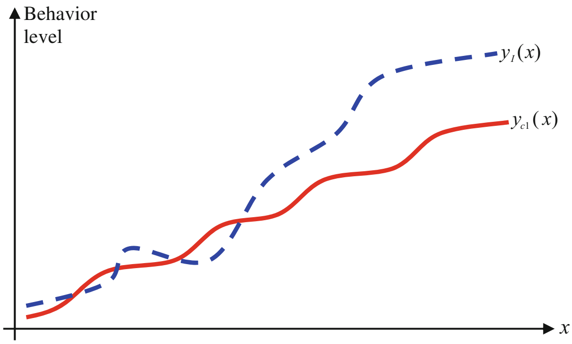


Fig. 4.13 One way to use the FLA for a social judgment

Fig. 4.14 Comparison of a type-1 consensus FLA behavior level and an individual's FLA behavior level



Their completed questionnaire can be interpreted as the individual's FLA, and its output can be plotted on the same plot as the output of the consensus FLA. These outputs can then be compared, and if some or all of the individual's outputs are "far" from those of the consensus FLA, then some action could be taken to sensitize the individual about these differences. Figure 4.14 depicts this for a type-1 consensus FLA.

One immediately sees a problem with the type-1 comparisons, namely, how "far" must the differences be between the individual FLA and the consensus FLA before some action (e.g., sensitivity training) is taken? This can be difficult to establish when one is comparing two functions, especially since "far" is in itself a

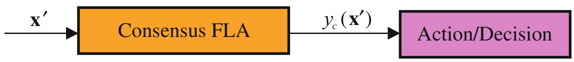


Fig. 4.15 Another way to use a FLA

fuzzy term. This problem is handled directly when type-2 fuzzy sets are used, and will be returned to in Chap. 10.

Another way to use a FLA is depicted in Fig. 4.15. After a consensus FLA has been designed, it is exposed to a situation, say $\mathbf{x} = \mathbf{x}'$, for which it provides the consensus output $y_c(\mathbf{x}')$. Then some action or decision occurs.

4.4.9 Connections to the Perceptual Computer

Since the first edition of this book (Mendel 2001), the author and his students, motivated by the FLA, developed a general methodology for assisting people in making subjective judgments, called *perceptual computing*. The *Perceptual Computer—Per-C*—is the instantiation of perceptual computing—and has the architecture that is depicted in Fig. 4.16 (Mendel 2001, 2002, 2007; Mendel and Wu 2010). It consists of three components: encoder, Computing With Words[17] (CWW) engine and decoder. Perceptions—words—activate the Per-C and are the Per-C output (along with data); so, it is possible for a human to interact with the Per-C using just a linguistic vocabulary.

A vocabulary is application (context) dependent, and must be large enough so that it lets the end-user interact with the Per-C in a user-friendly manner. The encoder transforms words into fuzzy sets and leads to a *codebook*—words with their associated fuzzy set models. The outputs of the encoder activate a CWW engine, whose output is one or more fuzzy sets, which are then mapped by the decoder into a recommendation (subjective judgment) with supporting data. The recommendation may be in the form of a word, group of similar words, rank or class.

---

[17]Zadeh (1996) coined the phrase *computing with words*, and stated: "CWW is a methodology in which the objects of computation are words and propositions drawn from a natural language. (It is) inspired by the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. CWW may have an important bearing on how humans … make perception-based rational decisions in an environment of imprecision, uncertainty and partial truth." He did not mean that computers would actually compute using words—single words or phrases—rather than numbers. He meant that computers would be activated by words, which would be converted into a mathematical representation using fuzzy sets, and that these fuzzy sets would be mapped by a CWW engine into some other fuzzy set, after which the latter would be converted back into a word. Zadeh's definition of CWW is very general and does not refer to a specific field in which CWW would be used.

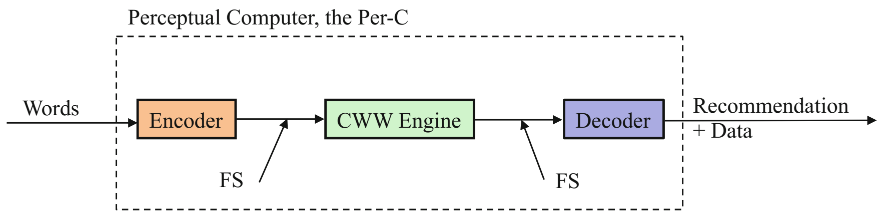


Fig. 4.16 Architecture for the perceptual computer

Although there are lots of details needed in order to implement the Per-C's three components—encoder, decoder and CWW engine [they are covered in Mendel and Wu (2010)]—it is when the Per-C is applied to specific applications, that the focus on the methodology becomes clear. Stepping back from those details, the *methodology of perceptual computing* is [adapted from Mendel and Wu (2010, p. 311)]:

1. Focus on an application (*A*), e.g. a specific social judgment.
2. Establish a vocabulary (or vocabularies) for *A*, e.g. the five-word vocabulary in Table 4.10.
3. Collect interval end-point data from a group of subjects (representative of the subjects who will use the Per-C) for all of the words in the vocabulary (see Sect. 5.2).
4. Map the collected word data into a fuzzy set model. The result of doing this is the *codebook* (or codebooks) for *A*, and completes the design of the encoder of the Per-C.
5. Choose an appropriate CWW engine for *A*. It will map fuzzy sets into one or more fuzzy sets. In this book only IF—THEN rules are used as a CWW engine. Other CWW engines include: Linguistic Weighted Average (Wu and Mendel 2007; Mendel and Wu 2010, Chap. 5) and Linguistic Weighted Power Mean (Rickard et al. 2011, 2013).
6. If an existing CWW engine is available for *A*, then use its available mathematics to compute its output(s). Otherwise, develop such mathematics for the new kind of CWW engine. The new CWW engine should be constrained[18] so that its output(s) resembles (is highly similar to) the fuzzy sets in the codebook(s) for *A*.
7. Map the fuzzy set outputs from the CWW engine into a recommendation at the output of the decoder. If the recommendation is a word, rank or class, then use existing mathematics to accomplish this mapping (Mendel and Wu 2010). Otherwise, develop such mathematics for the new kind of decoder.

Section 4.4.3 presented one way for mapping collected word data into a type-1 fuzzy set. It is a geometric construction that uses group statistics for each word.

[18]This (new) constraint is the major difference between perceptual computing and function approximation applications of fuzzy sets and systems.

Another approach is to fix the shape of the MF ahead of time, compute mathematical formulas for its (deterministic) mean and standard deviation, and then determine the MF parameter values by equating the data population mean and standard deviation to the MF mean and standard deviation. In order to obtain a unique solution from this approach each MF must be described by exactly two parameters. Detailed formulas can be found in Mendel and Wu (2010, Chap. 3, Tables 3.3 and 3.4).

Section 4.4.5 focused on a CWW Engine that averaged the responses.

No discussions have been given so far about the Decoder for the earlier FLAs. The obvious approach in going from $y_{c1}(x_1)$ (or $y_{c1}(x_2)$ or $y_{c1}(x_1, x_2)$) to a word is to locate its numerical value on the horizontal axis of Fig. 4.8, project this value up vertically so it intersects the word MFs, and choose the winning word as the one whose word MF has the largest numerical value (as explained in Sect. 2.2.4).

See Sect. 10.4.6 for further discussions about the Per-C when words are modeled as interval type-2 fuzzy sets.

4.5 Forecasting of Compressed Video Traffic[19] Using Mamdani and TSK Fuzzy Systems

Moving Picture Expert Group (MPEG) is a standard for digital video compression coding that has been extensively used to overcome the problem of storage of prerecorded video on digital storage media, because of the high compression ratios it achieves. MPEG video traffic is composed of a Group of Pictures (GoP) that contains the following encoded frames: intracoded (I), predicted (P), and bidirectional (B). I frames are coded with respect to the current frame using a two-dimensional discrete cosine transform; they have a relatively low compression ratio. P frames are coded with reference to previous I or P frames using interframe coding; they can achieve a better compression ratio than I frames. B frames are coded with reference to the next and previous I or P frame; they can achieve the highest compression ratio of the three frame types. The use of these three types of frames allows MPEG to be both robust (I frames permit error recovery) and efficient (B and P frames have high compression ratio).

Rose (1995) (University of Wurzburg), made 20 MPEG-1 video traces available on-line. He compressed the videos using an MPEG-1 encoder using a pattern with GoP size 12—IBBPBBPBBPBB. Observe that in this GoP there is one I frame, three P frames, and eight B frames. Each of his MPEG video streams consists of

[19]Although this application, which appears in Mendel (2001), may be dated, it has been left in this new edition because it is one that compares TSK and Mamdani fuzzy systems. For readers who are not interested in the discussions about compressed video, view Fig. 4.17 as another kind of time-series, where "time" is the same as "frame-index," and go directly to Sect. 4.5.1.

40,000 video frames (3333 I frames, 10,000 P frames, and 26,667 B frames), which, at 25 frames/s, represents about 30 mins of real-time full motion video.

Using these 20 compressed video traces, (Rose 1995) analyzed their statistical properties and observed that the frame and GoP sizes (i.e., bits per frame or bits per GoP) can be approximated by Gamma or Lognormal distributions. Additionally, Manzoni, et al. (1999) studied the workload models of variable bit rate (VBR) video traffic, Adas (1998) used adaptive linear prediction to forecast the VBR video for dynamic bandwidth allocation, and Krunz et al. (1995) found that the lognormal distribution is the best match for the frame sizes of all I/P/B frames; i.e., if the I, P, or B frame size at time j is s_j, then

$$\log_{10} s_j \approx N(\log_{10} s_j; m, \sigma^2) \tag{4.61}$$

Forecasting video traffic is important so that a telecommunication network can forecast its future traffic. By doing this, network bandwidth can be dynamically allocated.

4.5.1 Forecasting I Frame Sizes: General Information

The rest of this section focuses on the problem of forecasting I frame sizes (i.e., the number of bits/frame) for *Jurassic Park*, using TSK and Mamdani fuzzy systems. The methodologies for doing this apply as well to forecasting P and B frame sizes and can also be applied to other video products.

The following two designs of fuzzy system forecasters are examined, based on the logarithm of the first 1000 I frame sizes of *Jurassic Park*, $s(1)$, $s(2)$, …, $s(1000)$ (see Fig. 4.17): singleton type-1 TSK and Mamdani fuzzy systems. The first 504 data [$s(1)$, $s(2)$, …, $s(504)$] were used for tuning the parameters of these forecasters, and the remaining 496 data [$s(505)$, $s(502)$, …, $s(1000)$] were used for testing after tuning.

Singleton type-1 TSK fuzzy system: The rules of this fuzzy system forecaster are $(l = 1, \ldots, M)$

$$\begin{aligned} &R^l_{\text{TSK}}: \text{IF } s(k-3) \text{ is } F^l_1 \textit{ and } s(k-2) \text{ is } F^l_2 \text{ and } s(k-1) \text{ is } F^l_3 \textit{ and } s(k) \text{ is } F^l_4, \\ &\text{THEN } \hat{s}^l(k+1) = c^l_0 + c^l_1 s(k-3) + c^l_2 s(k-2) + c^l_3 s(k-1) + c^l_4 s(k) \end{aligned} \tag{4.62}$$

The F^l_j were chosen initially to be the same for all l and j, Gaussian MFs were used for them, and their initial means and standard deviations were computed from the first 500 I frames as $m = 4.73$ and $\sigma = 0.1$. According to Table 4.5 (the "TSK Singleton" column), the number of design parameters for this type-1 TSK fuzzy system is $(3p+1)M = 13M$. $\hat{s}^l(k+1)$ was computed by using (4.3) and the top

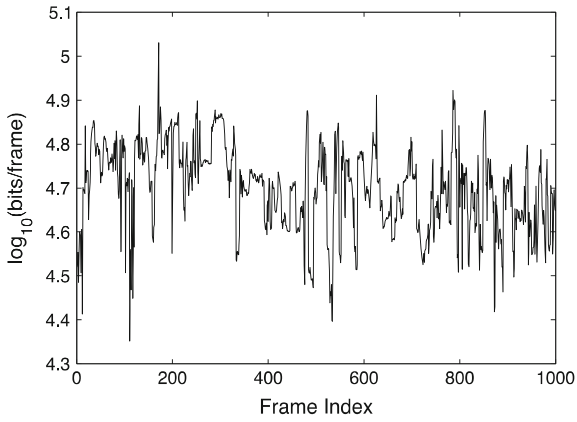


Fig. 4.17 The first 1000 I frame sizes of *Jurassic Park* MPEG-1 video data

line of (4.4) in which $y_{\text{TSK}}(\mathbf{x}) \equiv \hat{s}^l(k+1)$, $x_1 \equiv s(k-3)$, $x_2 \equiv s(k-2)$, $x_3 \equiv s(k-1)$ and $x_4 \equiv s(k)$.

Singleton type-1 Mamdani fuzzy system that uses height defuzzification: The rules of this fuzzy system forecaster are $(l = 1, \ldots, M)$

$$\begin{aligned} R_Z^l&: \text{IF } s(k-3) \text{ is } F_1^l \text{ and } s(k-2) \text{ is } F_2^l \text{ and } s(k-1) \text{ is } F_3^l \text{ and } s(k) \text{ is } F_4^l, \\ &\text{THEN } \hat{s}^l(k+1) \text{ is } G^l \end{aligned} \tag{4.63}$$

Height defuzzification was used, and, as was done for the singleton type-1 TSK fuzzy system, the F_j^l were chosen initially to be the same for all l and j, Gaussian MFs were used for them, and their initial means and standard deviations were also computed from the first 500 I frames as $m = 4.73$ and $\sigma = 0.1$. According to Table 4.5 (the "Mamdani Singleton" column), the number of design parameters for this type-1 singleton Mamdani fuzzy system is $(2p+1)M = 9M$. $\hat{s}^l(k+1)$ was computed by using (4.1) and the top line of (4.2) in which $y(\mathbf{x}) \equiv \hat{s}^l(k+1)$, $x_1 \equiv s(k-3)$, $x_2 \equiv s(k-2)$, $x_3 \equiv s(k-1)$, $x_4 \equiv s(k)$ and $\lambda^l = \bar{y}^l$.

4.5.2 *Forecasting I Frame Sizes: Using the Same Number of Rules*

In this first approach to designing the fuzzy system forecasters, the number of rules was fixed at five, i.e., $M = 5$. Doing this means that the singleton type-1 TSK fuzzy system is described by 65 design parameters, and the singleton type-1 Mamdani fuzzy system is described by 45 design parameters. Steepest descent algorithms (as described in Sect. 4.2.3) were used to tune all of these parameters, in which step sizes (learning parameters) of $\beta_\theta = 0.001$ and $\beta_\theta = 0.01$ were used for the TSK and Mamdani fuzzy systems, respectively.

How initial values were chosen for the MF parameters has already been explained. All of the remaining parameters were initialized randomly, as follows:

- Consequent parameters of the singleton TSK fuzzy system, $c_j^l \, (l = 1, \ldots, 5; \; j = 0, 1, \ldots, 4)$, were chosen randomly in [0, 0.2] with uniform distribution.
- Consequent parameters of the singleton Mamdani fuzzy system, $\bar{y}^i \, (i = 1, \ldots, 5)$, were chosen randomly in [0, 5] with uniform distribution.

Because the initial values of the consequent parameters were chosen randomly, 50 Monte Carlo realizations were run for each of the two designs. For each realization, each of the two fuzzy systems was tuned for 10 epochs on the 504 training data. The two designs were then evaluated on the remaining 496 testing data using the following RMSE:

$$\mathrm{RMSE} = \sqrt{\frac{1}{496} \sum_{k=504}^{999} \left[s(k+1) - y_{\mathrm{FLS}}(\mathbf{s}^{(k)}) \right]^2} \tag{4.64}$$

where $\mathbf{s}^{(k)} = [s(k-3), s(k-2), s(k-1), s(k)]^{\mathrm{T}}$ and $y_{\mathrm{FLS}} = y_{\mathrm{TSK}}$ or y. The average value and standard deviations of these RMSEs are plotted in Fig. 4.18 for each of the 10 epochs.

Observe, from Fig. 4.18a, that, after 10 epochs of training, the average RMSE of the two fuzzy system forecasters is: $\mathrm{Avg\,RMSE(TSK)} \approx 0.078$ and $\mathrm{Avg\,RMS(Mamdani)} \approx 0.08$; and, from Fig. 4.18b, that $\mathrm{Std\,RMSE(TSK)} \approx 0.009$ and $\mathrm{Std\,RMSE(Mamdani)} \approx 0.016$. Based on these numbers, because of the smaller standard deviation for the TSK fuzzy system, it seems that the singleton TSK fuzzy system gives somewhat better results than the singleton Mamdani fuzzy system.

4.5.3 Forecasting I Frame Sizes: Using the Same Number of Design Parameters

Because a five-rule singleton TSK fuzzy system always has more parameters (design degrees of freedom) to tune than does a comparable five-rule Mamdani fuzzy system, the previous approach to designing the two fuzzy systems was modified by fixing the rules used by the singleton TSK fuzzy system at five and by then choosing the number of rules used by the singleton Mamdani fuzzy system so that its total number of design parameters approximately equals the number for the TSK fuzzy system. Doing this led to using seven rules for the singleton Mamdani fuzzy system. The designs of the resulting two fuzzy systems proceeded exactly as described in the preceding section. All designs were again evaluated using the RMSE in (4.64). The average value and standard deviations of these RMSEs are plotted in Fig. 4.19 for each of the 10 epochs. Observe that: the results are similar to

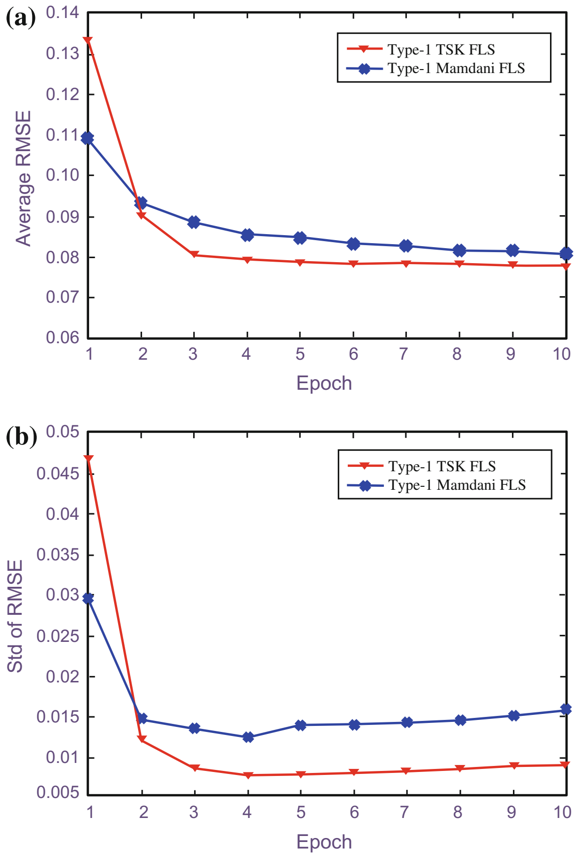


Fig. 4.18 The mean and standard deviations of RMSE (using the test data) for the two five-rule fuzzy system forecasters averaged over 50 Monte Carlo designs. Tuning was performed in each design for 10 epochs: **a** Mean values, and **b** standard deviations. Note that "FLS" is equivalent to "fuzzy system."

the ones depicted in Fig. 4.18, so, at least for this example, equalizing the numbers of design parameters in the Mamdani and TSK fuzzy systems does not seem to be so important.

4.6 Rule-Based Classification of Video Traffic

Pattern recognition usually involves (Duda 1994, p. 4)[20] "the extraction of significant characterizing features followed by classification on the basis of the values of the features." There are many approaches to pattern recognition, including deterministic, statistical, neural network, and, more recently, rule-based. This section explains the rule-based approach to a high-level video classification problem.

[20]This chapter was originally published in (Mendel and Fu 1970), and was reprinted in Mendel (1994), which was dedicated to K. S. Fu after his untimely death in the 1980s.

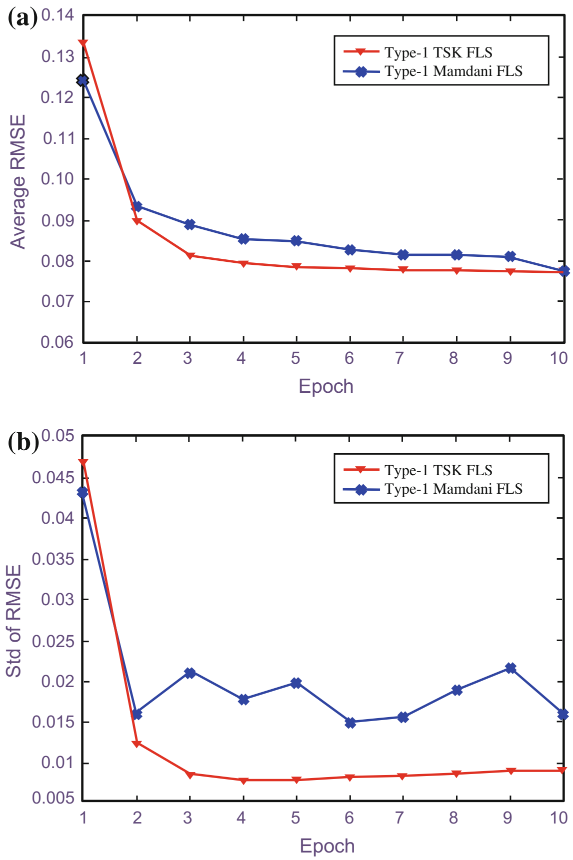


Fig. 4.19 The mean and standard deviations of RMSE (using the test data) for two fuzzy system forecasters averaged over 50 Monte Carlo designs. The singleton TSK and Mamdani fuzzy systems have approximately the same number of design parameters Tuning was performed in each design for 10 epochs: **a** Mean values, and **b** standard deviations. Note that “FLS” is equivalent to “fuzzy system.”

For a brief introduction to MPEG video traffic, see Sect. 4.5. The problem that is examined in the present section is the direct classification of compressed (MPEG-1) video traffic without decompressing it. This is for high-level classification, e.g. classify a video as a movie or a sports program, or as a movie or documentary. It is not for content-based classification, because content is part of a video frame (e.g., the gunfighter in movie A, the ballerina in movie B, the Eiffel Tower in movie C, etc.), whereas it is the entire frame that is available in compressed form. This section focuses on the two-class problem (i.e., classifying a video as either a movie or a sports program), leaving the more general case to the reader.

There can be two approaches to rule-based classification from compressed video traffic:

1. Decompress the video traffic, followed by classification
2. Classify the compressed video traffic directly

The first approach has the following disadvantages: it requires a decoder, which has a cost associated with it; decoding takes time, which introduces latency; and the decompressed video requires a lot of storage. The second approach—the one taken here—has the following advantages: it requires no decoder, so that no decoding time is needed and no additional storage is needed. Consequently, direct classification of compressed video traffic can save time and money.

Given a collection of MPEG-1 compressed movies and sports program videos, a subset of them are used to create (i.e., design and test) a rule-based classifier (RBC) in the framework of fuzzy systems. The overall approach taken here is to:

1. Choose appropriate features that act as the antecedents in a RBC
2. Establish the MFs for the features
3. Establish rules using the features
4. Optimize the rule design-parameters using a tuning procedure
5. Evaluate the performance of the optimized RBC using testing

The first three steps are relatively straightforward. The fourth step requires determining a computational formula for the RBC, in much the same way that such a formula was established earlier in this chapter for the Mamdani and TSK fuzzy systems. The fifth step requires base-lining the RBC; this is done using the accepted standard of a Bayesian classifier. All five steps are described in the rest of this section.

4.6.1 Selected Features

As is mentioned in Sect. 4.5, MPEG-1 compression produces three kinds of frames —I, P and B. Bits per I, P and B frames are available as measurements. The logarithm of bits per I-, P- and B-frames are used as the three features of the RBC. Figure 4.20 depicts these three kinds of frame sizes for the movie Terminator 2.[21] Observe that I frames have more bits/frame than P frames, which have more bits/frame than B frames.

4.6.2 MFs for the Features

Section 4.5 mentioned that (Krunz et al. 1995) found that the lognormal distribution is the best match for the frame sizes of all I/P/B frames. Additionally, Liang and Mendel (2001) showed[22] that the logarithms of I/P/B frame sizes are more

[21]This video was compressed using a GoP size 12—IBBPBBPBBPBB. A total of 40,000 video frames of Terminator 2 were available. To plot 3000 I, P, and B frame sizes, the first 36,000 frames were used for the I frame size plot in Fig. 4.20a; the first 12,000 frames were used for the P frame size plot in Fig. 4.20b; and, the first 4500 frames were used for the B frame size plot in Fig. 4.20c.

[22]See, also, Sect. 10.6.1.

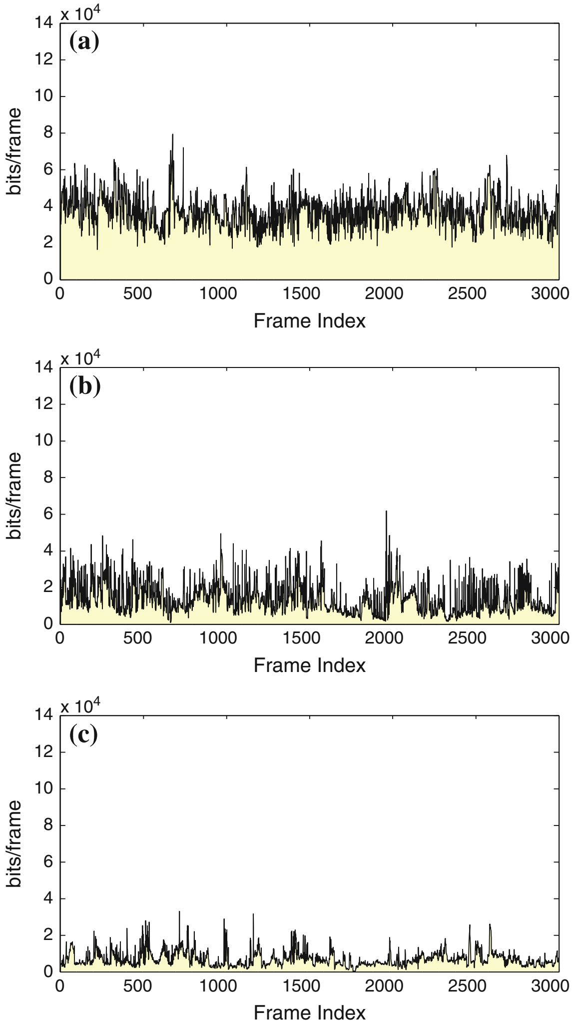


Fig. 4.20 Portions of I/P/B frame sizes of *Terminator 2* movie: **a** I frame, **b** P frame, and **c** B frame

appropriately modeled as Gaussians. Consequently, in this section Gaussian MFs are used for the features in the RBC.

4.6.3 Rules and Their Parameters

Rules for a RBC of compressed video traffic use the three selected features as their antecedents and have one consequent. The antecedents are: logarithm of bits/I frame, logarithm of bits/P frame, and logarithm of bits/B frame. The consequent is +1 if the video is a movie and −1 if it is a sports program. Observe that there is nothing fuzzy about a rule's consequent in rule-based classification; i.e., each rule's consequent is assigned a numerical value, +1 or −1.

Each rule in a *type-1* RBC has the following structure[23]:

$$R_Z^l\text{: IF I frame is } F_1^l \text{ and P frame is } F_2^l \text{ and B frame is } F_3^l\text{, THEN the video product is a movie } (+1) \text{ or a sports program } (-1) \tag{4.65}$$

Observe that this rule is a special case of a Zadeh rule, one in which the consequent is a singleton, i.e., (4.65) can be expressed as:

$$\begin{aligned} &R_Z^l\text{: IF I frame is } F_1^l \text{ and P frame is } F_2^l \text{ and B frame is } F_3^l\text{, THEN} \\ &y^l = \begin{cases} 1 & \text{for a movie} \\ -1 & \text{for a sports program} \end{cases} \end{aligned} \tag{4.66}$$

Only a very small number of rules were used in this study, namely one per video product, e.g. if the training set contains four movies and four sports programs, only eight rules are used.

Each Gaussian antecedent MF has two design parameters, its mean and standard deviation; hence, there are six design parameters per rule. Optimum values for all design parameters were determined during a tuning process; but, before such a process can be programmed, computational formulas for the RBC must be established.

4.6.4 Computational Formulas for the RBC

The MF $\mu_{G^l}(y)$ for the consequent of R^l in (4.66) can be expressed as $(l = 1, \ldots, M)$:

$$\mu_{G^l}(y) = \begin{cases} 1 & y = y^l \\ 0 & \text{otherwise} \end{cases} \tag{4.67}$$

Using (3.11), the MF of each three-antecedent fired rule, $\mu_{B^l}(y|\mathbf{x}')$, can be expressed as

$$\begin{aligned} \mu_{B^l}(y|\mathbf{x}') = \Bigg\{ &\left[\sup_{x_1 \in X_1} \mu_{X_1}(x_1|x_1') \bigstar \mu_{F_1^l}(x_1)\right] \bigstar \left[\sup_{x_2 \in X_2} \mu_{X_2}(x_2|x_2') \bigstar \mu_{F_2^l}(x_2)\right] \\ &\bigstar \left[\sup_{x_3 \in X_3} \mu_{X_3}(x_3|x_3') \bigstar \mu_{F_3^l}(x_3)\right] \Bigg\} \bigstar \mu_G^l(y) \end{aligned} \tag{4.68}$$

[23]See also Kuncheva (2000) for an excellent introduction to RBCs.

Applying (4.67) to this result, it follows that:

$$\mu_{B^l}(y|\mathbf{x}') = \begin{cases} \left\{ \begin{array}{l} \left[\sup_{x_1' \in X_1} \mu_{X_1}(x_1|x_1') \bigstar \mu_{F_1^l}(x_1)\right] \bigstar \left[\sup_{x_2 \in X_2} \mu_{X_2}(x_2|x_2') \bigstar \mu_{F_2^l}(x_2)\right] \\ \quad \bigstar \left[\sup_{x_3 \in X_3} \mu_{X_3}(x_3|x_3') \bigstar \mu_{F_3^l}(x_3)\right] \end{array} \right\} & y = y^l \\ 0 & y \neq y^l \end{cases} \tag{4.69}$$

For a *singleton type-1 RBC*, (4.69) reduces further to

$$\mu_{B^l}(y|\mathbf{x}') = \begin{cases} T_{k=1}^3 \mu_{F_k^l}(x_k') & y = y^l \\ 0 & y \neq y^l \end{cases} \equiv \begin{cases} f^l(\mathbf{x}') & y = y^l \\ 0 & y \neq y^l \end{cases} \tag{4.70}$$

Using height defuzzification [(3.39) and (3.40)], the output of a type-1 RBC can be expressed as

$$y_{\mathrm{RBC},1}(\mathbf{x}') = \frac{\sum_{l=1}^M f^l(\mathbf{x}') y^l}{\sum_{l=1}^M f^l(\mathbf{x}')} \tag{4.71}$$

in which $y^l = \pm 1$. The final decision that the measurements correspond to either a movie or a sports program is based on the sign of the defuzzified output, i.e.

$$\begin{cases} \mathrm{IF} y_{\mathrm{RBC},1}(x' > 0) & \text{decide movie} \\ \mathrm{IF} y_{\mathrm{RBC},1}(x' < 0) & \text{decide sports program} \end{cases} \tag{4.72}$$

If $y_{\mathrm{RBC},1}(\mathbf{x}') = 0$, flip a fair coin to decide whether the measurements correspond to a movie or a sports program.

Observe, from (4.71), that the normalization operation (i.e., $\sum_{l=1}^M f^l(\mathbf{x}')$) does not change the sign of $y_{\mathrm{RBC},1}(\mathbf{x}')$: hence, for two-category classification, (4.71) can be simplified. To that end, the following unnormalized output for the RBC in (4.72) is used:

$$y_{\mathrm{RBC},1}^U(\mathbf{x}') = \sum_{l=1}^M f^l(\mathbf{x}') y^l \tag{4.73}$$

4.6.5 Optimization of Rule Design Parameters

The simulation results to be discussed in Sect. 4.6.7 begin with five movies and five sports programs and, by way of illustration, are for fuzzy RBCs that use four movie rules and four sports program rules; i.e. each classifier has eight rules. The fuzzy RBC is optimized by using very simple modifications to the Sect. 4.2.3 steepest descent tuning procedure, modifications that are due to using an unnormalized output instead of a normalized output.

4.6.6 Testing the FL RBC

After the tuning of the FL RBC is completed, it is tested on the remaining unused products. Its false alarm rate[24] (FAR) is the performance measure that is used to evaluate it and to compare it against a Bayesian classifier, whose structure is described next.

It is well known that Bayesian decision theory [e.g. Duda et al. (2001)] provides the optimal solution to a general decision-making problem. To begin, assume that each video product (movie or sports program) v_i is given a numerical label (the five movies are labeled 1, 2, …, 5 and the five sports programs are labeled 6, 7, …, 10) and is equiprobable; i.e. $p(v_i) = 1/10$, where $i \in \{1, \ldots, 10\}$. Hypothesis H_1 is associated with a *movie*, whereas hypothesis H_2 is associated with a *sports program*, and $p(H_1) = p(H_2) = 0.5$.

It is also assumed that each component of the frame size vector $\mathbf{s}_i \equiv (s_i^I, s_i^P, s_i^B)^{\mathrm{T}}$ is a lognormal function of the I/P/B frames of the ith video product, and that $\mathbf{x}_i \equiv \log \mathbf{s}_i$; hence, $(i = 1, \ldots, 10)$

$$p(\mathbf{x}_i|v_i) = \frac{1}{(2\pi)^{3/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{m}_i)^{\mathrm{T}}\Sigma_i^{-1}(\mathbf{x}_i - \mathbf{m}_i)\right] \tag{4.74}$$

where $\mathbf{m}_i \equiv (m_i^I, m_i^P, m_i^B)^{\mathrm{T}}$ and $\Sigma_i \equiv \mathrm{diag}(\sigma_i^{I^2}, \sigma_i^{P^2}, \sigma_i^{B^2})$ are the mean vector (3×1) and covariance matrix (3×3) of $\mathbf{x}_i$, respectively, and $\mathbf{m}_i$ and Σ_i are estimated from the training data.

In this case,

$$p(\mathbf{x}|H_1) = \sum_{i=1}^{5} p(\mathbf{x}|v_i)p(v_i) \tag{4.75}$$

and

$$p(\mathbf{x}|H_2) = \sum_{i=6}^{10} p(\mathbf{x}|v_i)p(v_i) \tag{4.76}$$

[24] A *false alarm* occurs when an incorrect decision is made, i.e., when the classifier decides the video is a movie when it actually is a sports program, or when it decides the video is a sports program when it actually is a movie.

Because $p(H_1) = p(H_2) = 0.5$, the Bayes decision rule is:

$$\begin{array}{ll} \text{The video traffic is a movie if} & p(\mathbf{x}|H_1) > p(\mathbf{x}|H_2) \\ \text{The video is a sports program if} & p(\mathbf{x}|H_1) < p(\mathbf{x}|H_2) \\ \text{The video is a movie or a sports program if} & p(\mathbf{x}|H_1) = p(\mathbf{x}|H_2) \end{array} \tag{4.77}$$

4.6.7 Results and Conclusions

So as not to get lost in the details associated with the design and testing of the fuzzy RBC, the reader is referred to Liang and Mendel (2001) for them. Here the focus is on one set of results for so-called *out-of-product classification*. "Out-of-product" means that *some* of the compressed data are used from *some* of the *available* video products to establish the rules and to optimize (tune) the resulting classifier, after which the classifier is tested on the unused video products. As mentioned earlier, eight video products (four movies and four sports programs) out of a total of 10 available products were used to design the RBC. The first 24,000 (out of 40,000) compressed frames of each of the eight video products were used to establish and design an eight-rule fuzzy RBC. The first 37,500 compressed frames of the remaining two videos were then used for testing. An exhaustive study of the 25 possible designs was conducted. Average FAR (averaged aver the 25 possible designs) for the fuzzy RBC as well as for the Bayesian classifier are[25]:

- Singleton type-1 fuzzy RBC: FAR = 9.41%
- Bayesian classifier: FAR = 14.29%

From these results, observe that the type-1 singleton fuzzy RBC provides the best performance, and has 34.15% fewer false alarms than does the Bayesian classifier. Additional simulation studies that use 20 video products (10 movies and 10 sports programs) have also been performed and support these conclusions.

In summary, it has been demonstrated that it is indeed possible to perform high-level classification of movies and sports programs working directly with compressed data. This study is returned to again in Sect. 10.6, where even better performance is obtained by using interval type-2 fuzzy sets instead of type-1 fuzzy sets.

[25]The first edition of this book Mendel (2001) included results for a non-singleton type-1 fuzzy RBC. Because it is known that for Gaussian MFs, the final tuned results for non-singleton and singleton Mamdani fuzzy systems are the same (e.g., see Sect. 4.3.6 and the end of Example 3.7), the non-singleton fuzzy RBC has not been included in this 2nd edition.

4.7 Case Study: Fuzzy Logic Control

This section provides a short early history of fuzzy control, explains what a type-1 fuzzy logic controller is, and provides: a short introduction to fuzzy PID control, the general structure of a fuzzy PID controller, conventional and fuzzy PID control design methods, and some simulation results comparing T1 fuzzy PID control with PID control.

4.7.1 Early History of Fuzzy Control[26]

Fuzzy control (also known as fuzzy logic control) is regarded as the most widely used application of fuzzy logic and is credited with being a well-accepted methodology for designing controllers that are able to deliver satisfactory performance in the face of uncertainty and imprecision (Lee 1990; Sugeno 1985; Feng 2006). In addition, fuzzy logic theory (using engineering implications) provides a method for less skilled personnel to develop practical control algorithms in a user-friendly way that is close to human thinking and perception, and to do this in a short amount of time. Fuzzy logic controllers (FLCs) can sometimes outperform traditional control systems (like PID controllers) and have often performed either similarly or even better than human operators. This is partially because most FLCs are nonlinear controllers that are capable of controlling real-world systems (the vast majority of such systems are nonlinear) better than a linear controller can, and with minimal to no knowledge about the mathematical model of the plant or process being controlled.

FLCs have been applied with great success to many real-world applications. The first FLC was developed in Mamdani and Assilian (1975), in the United Kingdom, for controlling a steam generator in a laboratory setting. In 1976, Blue Circle Cement and SIRA in Denmark developed a cement kiln controller (the first industrial application of fuzzy logic), which went into operation in 1982 (Holmblad and Ostergaard 1982). In the 1980s, several important industrial applications of fuzzy logic control were launched successfully in Japan, including a water-treatment system developed by Fuji Electric. In 1987, Hitachi put a fuzzy logic-based automatic train operation control system into the Sendai city's subway system (Yasunobu and Miyamoto 1985). These and other applications of FLCs motivated many Japanese engineers to investigate a wide range of novel applications for fuzzy logic. This led to a "fuzzy boom" in Japan, a result of close collaboration and technology transfer between universities and industry.

According to Yen and Langari (1999), in 1988, a large scale national research initiative was established by the Japanese Ministry of International Trade and Industry (MITI). The initiative established by MITI was a consortium called the

[26]The material in this section is taken from Mendel et al. (2014, Chap. 1).

Laboratory for International Fuzzy Engineering Research (LIFE). In late January 1990, Matsushita Electric Industrial (Panasonic) named their newly developed fuzzy controlled automatic washing machine, the fuzzy washing machine and launched a major commercial campaign of it as a *fuzzy* product. This campaign turned out to be a successful marketing effort not only for the product, but also for fuzzy logic technology (Yen and Langari 1999). Many other home electronics companies followed Panasonic's approach and introduced fuzzy vacuum cleaners, fuzzy rice cookers, fuzzy refrigerators, fuzzy camcorders (for stabilizing the image under hand jittering), fuzzy camera (for smart autofocus), and others. As a result, consumers in Japan recognized the now en-vogue Japanese word "fuzzy," which won the gold prize for a new word in 1990 (Hirota 1995). Originating in Japan, the "fuzzy boom" triggered a broad and serious interest in this technology in Korea, Europe, the USA and elsewhere. For example, Boeing, NASA, United Technologies and other aerospace companies developed FLCs for space and aviation applications (Munakata and Jani 1994).

Today FLCs are used in countless real-world applications that touch the lives of people all over the world, including white goods [e.g., washing machines, refrigerators (fridges), microwaves, rice cookers, televisions, etc.], digital video cameras, cars, elevators (lifts), heavy industries (e.g., cement, petroleum, steel), etc.

4.7.2 What Is a Type-1 Fuzzy Logic Controller (FLC)?[27]

Fuzzy logic control aims to mimic the process followed by the human mind when performing control actions. For example, when a person drives (controls) a car, he/she will not think:

> If the temperature is 10 °C and the rainfall is 70.5 mm and the road is 40% *slippery* and the distance between my car and the car in front of me is 3 m, then I will depress the acceleration pedal only 10%.

Instead, it is much more likely that he/she thinks:

> If it is *cold* and the rainfall is *high* and the road is *somewhat slippery* and the distance between my car and the car in front of me is *quite close*, then I will depress the acceleration pedal *slightly*.

So, in systems controlled by humans, the control cycle starts by a person converting a physical quantity (e.g., a distance) from numbers into words or perceptions (e.g., *quite close* distance). The input words (or perceptions) then trigger a person's knowledge, accumulated through that person's experience, resulting in words representing actions (e.g., depress the acceleration pedal *slightly*). The person then executes an action to actuate a given device that interfaces the person with the controlled system (e.g., depress the acceleration pedal only 10% might represent the

[27]The material in this section is taken from Mendel et al. (2014, Chap. 1).

person's implementation of "depress the accelerator pedal *slightly*"). Because people think and reason by using imprecise linguistic information, FLCs try to mimic and convert linguistic control information into numerical control information that can be used in automatic control systems.

In an attempt to mimic human control actions, a type-1 FLC has the structure of the fuzzy system in Fig. 3.1.

In general, real-world control systems, including fuzzy logic control systems, are affected by the following uncertainties:

- Uncertainties about the inputs to the FLC. For instance, sensor measurements can be affected by high noise levels and changing observation conditions such as changing environmental conditions, e.g., wind, rain, humidity, etc. In addition to measurement noise, other possible inputs to the FLC, such as those estimated by an observer or computed using a process model, can also be imprecise and exhibit uncertainty.
- Uncertainties about control outputs that can occur because of changes in an actuator's characteristics due to wear and tear, environmental changes, etc.
- Uncertainties about the change in operating conditions of the controller, such as changes in a plant's parameters.
- Uncertainties due to disturbances acting upon the system when those disturbances cannot be measured, e.g., wind buffeting an airplane.

In a type-1 FLC all of these uncertainties are handled by the type-1 fuzzy sets in the antecedents and consequents of the rules, as well as through the chosen type of fuzzifier.

The type-1 FLC in Fig. 3.1 is a nonlinear controller that maps its inputs $\mathbf{x}'$ into an output u, i.e., $u = f(\mathbf{x}')$, where f is a nonlinear function that is formed by fuzzy logic operations and the mathematics of fuzzy sets. Often, $f(\mathbf{x}')$ is formed from linguistic rules that summarize human knowledge or experience (or may be constructed from data); thus, the type-1 FLC directly maps such knowledge or experience into a nonlinear control law. This has enabled fuzzy logic control to be used in complex ill-defined processes, especially those that can be controlled by a skilled human operator without the knowledge of their underlying dynamics (Mamdani and Assilian 1975).

Recall that *Variable Structure Control* is a form of discontinuous nonlinear control that alters the dynamics of a nonlinear system through the application of high-frequency switching control. A type-1 FLC can also be regarded as variable-structure controller by virtue of the mathematics of fuzzy sets and systems, i.e. it partitions the state space *automatically* rather than by a planned design. This is because different rules are activated for different regions of the state space. Palm (1992) showed that a FLC can be regarded as an extension of a conventional variable structure controller with a boundary layer.

Mamdani and TSK fuzzy systems have both been used as FLCs. According to Mamdani (1994): When tuned, the parameters of a PID controller affect the shape of the entire control surface. Because fuzzy logic control is a rule-based controller, the shape of the control surface can be individually manipulated for the different regions of the state space, thus limiting possible effects only to neighboring regions (see, e.g., Example 3.3 and Fig. 3.3).

A FLC can be studied like any other nonlinear controller, e.g., for the Mamdani FLC, stability and robustness studies can be performed by extensive simulations and by analyzing its control surface. For a TSK FLC, it is possible to perform the same kinds of mathematical analyses that are applied to other nonlinear controllers, such as Lyapunov stability and robustness, etc.

Many details about fuzzy logic control are in, e.g., Feng (2006), Ying (2000), Mendel et al. (2014).

4.7.3 Fuzzy PID Control[28]

4.7.3.1 Background

It is a known fact that conventional PID controllers are the most popular controllers used in industry due to their simple structure and cost efficiency [e.g., Astrom and Hagglund (2001, 2005), Skogestad (2003), Seborg et al. (2004)]. However, a linear PID controller is not well suited for strongly nonlinear and uncertain systems. Fuzzy PID controllers (FPID) are often mentioned as an alternative to conventional PID controllers since they are analogous to the conventional PID controllers from an input–output relationship point of view [e.g., Galichet and Foulloy (1995), Moon (1995)]. After the pioneering study in Qiao and Mizumoto (1996) numerous techniques were developed for analyzing and designing a wide variety of FPID control systems [e.g., Li and Gatland (1996), Huang et al. (1999), Hu et al. (2001), Duan et al. (2008), Kumbasar and Hagras (2015)].

4.7.3.2 General Structure of Fuzzy PID Controller

A two-input direct action type FPID controller is depicted in Fig. 4.21 (Galichet and Foulloy 1995; Moon 1995; Qiao and Mizumoto 1996; Li and Gatland 1996; Huang et al. 1999; Hu et al. 2001; Duan et al. 2008; Kumbasar and Hagras 2015). The output of the standard FPID controller (u) is constructed by choosing the inputs to be error (e) and derivative of error ($\dot{e}$). The FPID controller structure uses scale factors (SFs) for its two inputs and outputs. K_e and K_d normalize the inputs e and $\dot{e}$ to the common interval $[-1,1]$; thus, e and $\dot{e}$ are converted, after normalization, into E and $\dot{E} \equiv \Delta E$, respectively. The FLC output (U) is mapped into its actual output (u) using the output SFs K_0 and K_1.

[28]This section was prepared by Prof. Tufan Kumbasar.

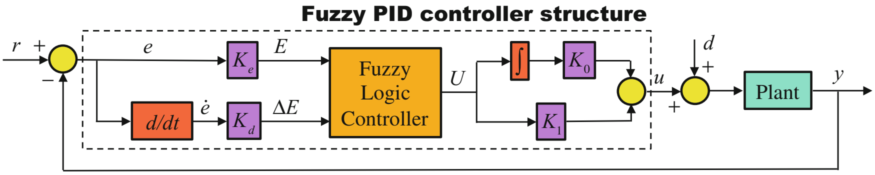


Fig. 4.21 Illustration of the FPID controller structure

Table 4.14 The rule base of the FPID controller

E/ΔE	N	Z	P
N	$R_Z^1: U = NB \equiv -1$	$R_Z^2: U = NM \equiv -0.5$	$R_Z^3: U = Z \equiv 0$
Z	$R_Z^4: U = NM \equiv -0.5$	$R_Z^5: U = Z \equiv 0$	$R_Z^6: U = PM \equiv 0.5$
P	$R_Z^7: U = Z \equiv 0$	$R_Z^8: U = PM \equiv 0.5$	$R_Z^9: U = PB \equiv 1$

The FPID controller considered here uses the Table 4.14 symmetrical 3×3 rule base, in which $N =$ negative, $Z =$ zero, $P =$ positive, $NB =$ negative big, $NM =$ negative medium, $PM =$ positive medium and $PM =$ positive big. The rule structure of the FLC is $(l = 1, \ldots, 9)$:

$$R_Z^l : \text{IF } E \text{ is } F_1^l \text{ and } \Delta E \text{ is } F_2^l \text{ THEN } U \text{ is } G^l \tag{4.78}$$

In (4.78), both E and ΔE are described by the three 50% overlapping triangle MFs that are depicted in Fig. 4.22a, and G^l are crisp singletons that are shown in Fig. 4.22b and are tabulated in Table 4.14.

A Mamdani FPID controller was implemented using singleton fuzzification, product implication and COS defuzzification (Exercise 4.21).

4.7.3.3 Conventional and Fuzzy PID Controller Design Methods

It is well known that a large number of high-order or nonlinear industrial plants can be modeled by the following first-order plus time delay (FOPTD) transfer function:

$$G(s) = \frac{K}{Ts + 1} e^{-Ls} \tag{4.79}$$

where K, T and L are the gain, time constant and time delay, respectively. The wide use of this plant model is due to both its simplicity and its ability to capture the essential dynamics of several industrial processes [e.g., Skogestad (2003), Seborg et al. (2005), Astrom and Hagglund (2005)]. Moreover, using this plant model provides one with the opportunity to design a PID controller using the Internal Model Control (IMC) principle (Francis and Wonham 1976), in which (Skogestad 2003; Seborg et al. 2005) the PID structure includes a low-pass filter, and is given by the following transfer function:

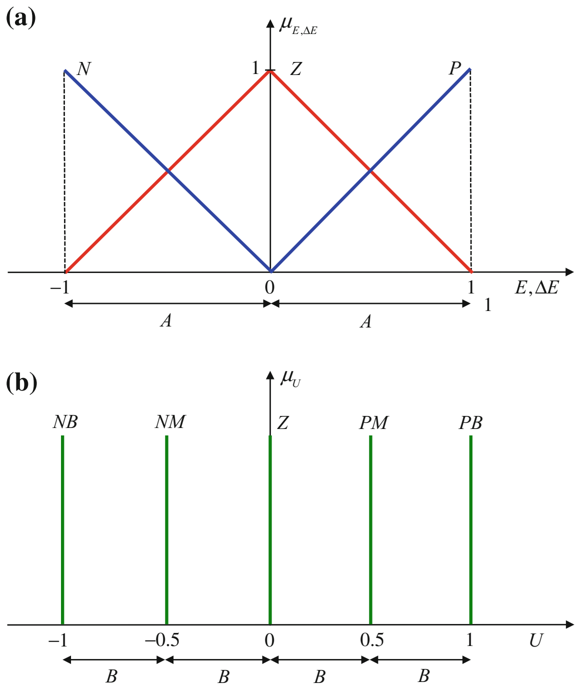


Fig. 4.22 **a** Antecedent MFs of the FPID controller, and **b** consequent values

$$\mathrm{PID(s)} = K_c\left(1 + \frac{1}{\tau_I s} + \tau_D s\right)\left(\frac{1}{1 + \tau_f s}\right) \tag{4.80}$$

In (4.80), K_c is a proportional gain, τ_I is the integral time constant, τ_D is the derivative time constant and τ_f is the low-pass filter time constant. These four parameters are defined as:

$$\begin{cases} K_c \equiv \dfrac{T + 0.5L}{K(\tau_c + L)} \\ \tau_I \equiv T + 0.5L \\ \tau_D \equiv \dfrac{0.5TL}{T + 0.5L} \\ \tau_f \equiv \dfrac{0.5L\tau_c}{\tau_c + L} \end{cases} \tag{4.81}$$

where τ_c is the desired closed-loop time constant and is the only design parameter to be tuned (Skogestad 2003; Seborg et al. 2005). In this study it was tuned to a value of $\tau_c = 0.2$ so as to provide the system with a fast and satisfactory response.

The IMC-based design strategy has been extended to FPID controllers to determine the scale-factors based on FOPDT model approximation. Duan et al. (2008) have shown that scale-factors K_d, K_0 and K_1 can be calculated as:

$$\begin{cases} K_d = \min(T, L/2) \times K_e \\ K_0 = \dfrac{A}{B} \times \dfrac{1}{KK_e(\tau_c + L/2)} \\ K_1 = \max(T, L/2) \times K_0 \end{cases} \tag{4.82}$$

In (4.82), A and B are the half-spreads of the input and output MFs (see Fig. 4.22). The remaining tuning parameter for the FPID controller is also τ_c, which was set to $\tau_c = 0.2$ in order to show the effects of using fuzzy sets [type-1 and also interval type-2 (Sect. 10.8)]. Finally, the input scale factor K_e is defined here as:

$$K_e \equiv \frac{1}{r(t_f) - y(t_f)} \tag{4.83}$$

In (4.83), $r(t_f)$ and $y(t_f)$ are the values of the reference and system output, respectively, at the time instant $(t = t_f)$ at which the reference varies.

4.7.3.4 Simulation Results (T1-FPID Versus PID):

This section compares the performances of the IMC based T1-FPID and PID controllers that were designed for a FOPTD *Nominal Process*, whose parameters are: $K = 1$, $L = 1$ and $T = 1$. Additionally, in order to examine the robustness of the two controllers, the following perturbed processes were considered:

- *Perturbed Process-1* : $K = 1.4$, $L = 1.2$ and $T = 1.2$
- *Perturbed Process-2* : $K = 0.6$, $L = 2$ and $T = 0.9$

Step responses for three situations are depicted in Figs. 4.23, 4.24 and 4.25. % overshoot (OS), settling time (T_s) and integral absolute error (IAE) values are given in Table 4.15. As can be clearly seen, the IMC based T1-FPID controller produces superior control performance as compared with its conventional PID controller counterpart. For example, for the Nominal Process the % overshoot for the T1-FPID controller is 0% whereas it is 28.1% for the PID controller, and the settling time for the T1-FPID controller is 4.8 s whereas it is 6.3 s for the PID controller. It is also very clear from Figs. 4.24 and 4.25 that the T1-FPID controller is much more robust to changes in the Nominal Process, as is evidenced by its much less oscillatory step response behavior. This robustness is supported by the OS, T_s and IAE values that are given in Table 4.15.

In summary, the T1-FPID was easy to design and gave significantly better performance than the conventional PID controller. Of course, the performance of the conventional PID controller can be enhanced by a prediction mechanism, cascaded PID, etc. However, the performance of the T1-FPID controller can also be enhanced by using 5×5 or 7×7 rule bases, and by optimizing the MF parameters and/or scaling factors.

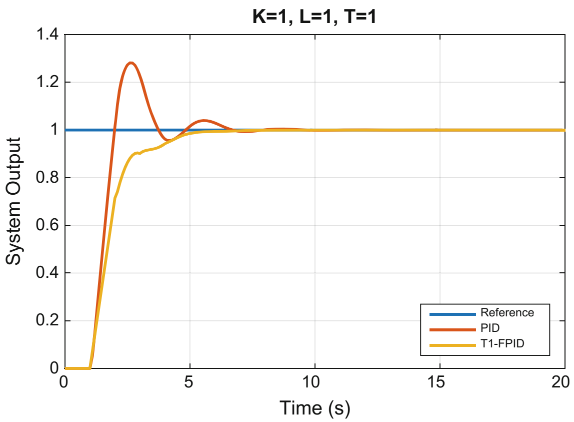


Fig. 4.23 System step responses for the Nominal Process

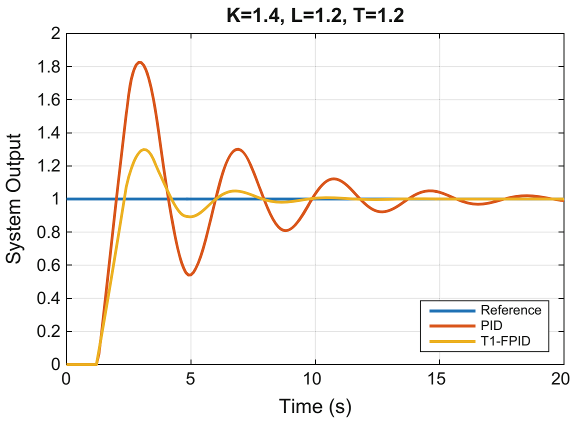


Fig. 4.24 System step responses for the Perturbed Process-1

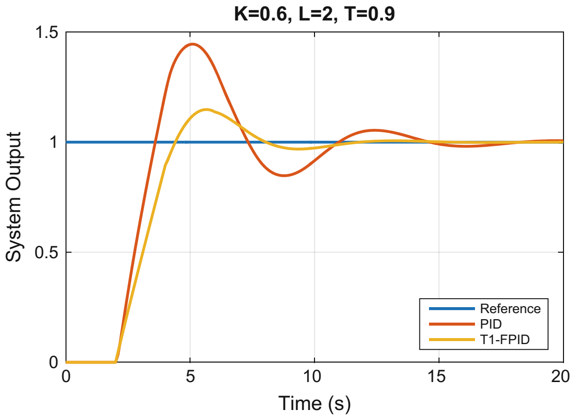


Fig. 4.25 System step responses for the Perturbed Process-2

Table 4.15 Control performance comparison of the FPID controllers

Controller	Nominal process			Perturbed process-1			Perturbed process-2		
	OS (%)	T_s(s)	IAE	OS (%)	T_s(s)	IAE	OS (%)	T_s(s)	IAE
PID	28.1	6.3	19.5	84.3	18.6	43.1	43.6	14.0	43.9
T1-FPID	0.0	4.8	19.8	29.9	7.6	23.9	14.7	10.5	35.6

Appendix 1: Proof of Theorem 4.1

The goal here is to show that, because the QPSO algorithm retains the best particle, this particle must have a performance that is at least as good as that of the optimized singleton fuzzy system. To demonstrate that this is true *the pseudocode in* Table 4.6 *is used* in which the first particle for the optimized design of a non-singleton fuzzy system is the optimized singleton fuzzy system (i.e., $\mathbf{p}_1(1) = \boldsymbol{\theta}_1(1) = \boldsymbol{\theta}_{\text{STI}}$) and the remaining $N_m - 1$ particles are chosen randomly for the non-singleton fuzzy system. The approach here is to go through the entire pseudocode two times in order to clearly demonstrate how or if the singleton fuzzy system particle can survive the QPSO iterations. The rest of this appendix follows (Mendel 2014, Appendix A) very closely.

QPSO Iteration 1 $(t = 1)$

$\mathbf{m}(1)$ is computed, as

$$\mathbf{m}(1) = \frac{1}{N_m}\left[\boldsymbol{\theta}_{\text{STI}} + \sum_{m=2}^{N_m} \mathbf{p}_m(1)\right] \tag{4.84}$$

then, $J(\mathbf{p}_m(1))$ is computed for $m = 1, \ldots, N_m$, after which $\mathbf{p}_{\text{gbest}}(1)$ is computed, as

$$\mathbf{p}_{\text{gbest}}(1) = \arg \min_{\mathbf{p}_m(1), \forall m=1,\ldots,N_m} J(\mathbf{p}_m(1)) \tag{4.85}$$

Two cases are now possible: $\mathbf{p}_{\text{gbest}}(1) = \mathbf{p}_1(1) = \boldsymbol{\theta}_{\text{STI}}$ or $\mathbf{p}_{\text{gbest}}(1) \neq \mathbf{p}_1(1)$.

(i) $\mathbf{p}_{\text{gbest}}(1) = \mathbf{p}_1(1) = \boldsymbol{\theta}_{\text{STI}}$: For $t = 1$ it is then always true, for *all particles*, that $J(\boldsymbol{\theta}_m(1)) = J(\mathbf{p}_m(1))$ because $\mathbf{p}_m(1) = \boldsymbol{\theta}_m(1)$; hence, no changes are made as a result of the test "If $J(\boldsymbol{\theta}_m(1)) < J(\mathbf{p}_m(1))$, then $\mathbf{p}_m(1) = \boldsymbol{\theta}_m(1)$". For each component of *Particle 1* $(m = 1)$, regardless of the value chosen for η, it is therefore true that $(j = 1, \ldots, N_{\boldsymbol{\theta}})$:

$$\begin{aligned} p_{1,j}(2) &= \eta \times p_{1,j}(1) + (1-\eta) \times p_{\text{gbest},j}(1) \\ &= \eta \times p_{1,j}(1) + (1-\eta) \times p_{1,j}(1) = p_{1,j}(1) = \boldsymbol{\theta}_{\text{STI},j} \end{aligned} \tag{4.86}$$

This means that $\mathbf{p}_1(2) = \boldsymbol{\theta}_{\text{STI}}$.

Note, also that: (1) the **p**-vectors for particles $m = 2, \ldots, N_m$ change from $\mathbf{p}_m(1)$ to $\mathbf{p}_m(2)$ by blending $\mathbf{p}_m(1)$ and $\mathbf{p}_{\text{gbest}}(1)$ according to $p_{m,j}(2) = \eta p_{m,j}(1) + (1-\eta)\ \boldsymbol{\theta}_{\text{STI},j}$, and $p_{m,j}(2) \neq \boldsymbol{\theta}_{\text{ST1}}$, since, for these values of m, $p_{m,j}(1) \neq \boldsymbol{\theta}_{\text{ST1},t}$; and, (2) another N_m $\boldsymbol{\theta}$-particles are created by making use of both $\mathbf{m}(1)$ and the just-computed $\mathbf{p}_m(2)$ (this is also done for Particle 1).

So, when $\mathbf{p}_{\text{gbest}}(1) = \mathbf{p}_1(1) = \boldsymbol{\theta}_{\text{STI}}$ the optimized singleton type-1 particle survives to be used during the next iteration of QPSO.

(ii) $\mathbf{p}_{\text{gbest}}(1) \neq \mathbf{p}_1(1)$: For $t = 1$, it is still always true, for all particles, that $J(\boldsymbol{\theta}_m(1)) = J(\mathbf{p}_m(1))$ because $\mathbf{p}_m(1) = \boldsymbol{\theta}_m(1)$; hence, again no changes are made as a result of the test "If $J(\boldsymbol{\theta}_m(1)) < J(\mathbf{p}_m(1))$, then $\mathbf{p}_m(1) = \boldsymbol{\theta}_m(1)$". Now, however, for *Particle 1* $(m = 1)$

$$p_{1,j}(2) = \eta \times p_{1,j}(1) + (1-\eta) \times p_{\text{gbest},j}(1) \neq p_{1,t}(1) = \boldsymbol{\theta}_{\text{STI},j} \tag{4.87}$$

This means the optimized singleton fuzzy system particle does not survive to be used during the next iteration of QPSO, i.e. it is the end of the line for the optimized singleton fuzzy system, and the performance of the non-singleton fuzzy system is already better than that of the optimized singleton fuzzy system.

QPSO Iteration 2 $(t = 2)$

$\mathbf{m}(2)$ is computed, as

$$\mathbf{m}(2) = \begin{cases} \frac{1}{N_m}\left[\boldsymbol{\theta}_{\text{STI}} + \sum_{m=2}^{N_m} \mathbf{p}_m(2)\right] & \text{If } \mathbf{p}_{\text{gbest}}(1) = \mathbf{p}_1(1) = \boldsymbol{\theta}_{\text{STI}} \\ \frac{1}{N_m}\sum_{m=1}^{N_m} \mathbf{p}_m(2) & \text{If } \mathbf{p}_{\text{gbest}}(1) \neq \mathbf{p}_1(1) \end{cases} \tag{4.88}$$

It is only necessary to focus on the results from iteration 1 of QPSO for which the optimized singleton fuzzy system particle survived to see how or if it can survive this second iteration of QPSO. Again, two cases are possible: (1) $\mathbf{p}_{\text{gbest}}(2) = \mathbf{p}_1(2) = \boldsymbol{\theta}_{\text{STI}}$ or (2) $\mathbf{p}_{\text{gbest}}(2) \neq \mathbf{p}_1(2)$.

(i) $\mathbf{p}_{\text{gbest}}(2) = \mathbf{p}_1(2) = \boldsymbol{\theta}_{\text{STI}}$: For $t = 2$ the optimized singleton fuzzy system particle survives only if, in addition to $\mathbf{p}_{\text{gbest}}(2) = \mathbf{p}_1(2) = \boldsymbol{\theta}_{\text{STI}}$, it is also true that $J(\boldsymbol{\theta}_1(2)) \not< J(\mathbf{p}_1(2)) = J(\boldsymbol{\theta}_{\text{STI}})$. It is conceivable that this can occur, in which case:

$$\begin{aligned} p_{1,j}(3) &= \eta \times p_{1,j}(2) + (1-\eta) \times p_{\text{gbest},j}(2) \\ &= \eta \times p_{1,j}(2) + (1-\eta) \times p_{1,j}(2) = p_{1,j}(2) = \boldsymbol{\theta}_{\text{STI},j} \end{aligned} \tag{4.89}$$

This means that $\mathbf{p}_1(3) = \boldsymbol{\theta}_{STI}$. So, when $\mathbf{p}_{gbest}(2) = \mathbf{p}_1(2) = \boldsymbol{\theta}_{STI}$ it is indeed possible for the optimized singleton fuzzy system particle 1 to survive to be used during the next iteration of QPSO, although it is more difficult for it to do so in this second iteration of QPSO because for it to happen it must survive two tests.

(ii) $\mathbf{p}_{gbest}(2) \neq \mathbf{p}_1(2)$: In this case, regardless of whether or not the test "If $J(\boldsymbol{\theta}_m(2)) < J(\mathbf{p}_m(2))$" is passed, none of the $\mathbf{p}$- or $\boldsymbol{\theta}$-particles will be the same as the optimized singleton fuzzy system particle; hence, the optimized singleton fuzzy system particle again does not survive to be used during the next iteration of QPSO, i.e. it is the end of the line for the optimized singleton fuzzy system particle, and the performance of the non-singleton fuzzy system particle is now better than that of the optimized singleton fuzzy system.

Further QPSO Iterations

It should be clear, from the above details of iterations 1 and 2 of QPSO that it is extremely difficult for the optimized singleton fuzzy system particle to continue to survive. It may happen, in which case the performance of the non-singleton fuzzy system is that of the singleton fuzzy system; however, it is highly unlikely to happen especially if the number of iterations of QPSO is made large enough.

It appears, from the above brief analysis of the QPSO algorithm that it minimizes $J(\boldsymbol{\theta})$ in a monotonically nonincreasing manner.

Exercises

4.1 Suppose that triangles are used for all interior MFs and piecewise linear functions are used for the two shoulder (exterior) MFs. Redo Tables 4.4 and 4.5 for these choices.

4.2 Towards the end of Sect. 4.1 four designs were stated for Mamdani and TSK *non-singleton* fuzzy systems (e.g., Fix the shapes and parameters of all the antecedent, consequent, and input measurement MFs ahead of time. The data establish the rules and the standard deviation of the measurements, and no tuning is used.). Explain how to use one of the design methods that are described in Sect. 4.2 for each of these four designs.

4.3 Repeat Example 4.4 for the six sampled data points that are depicted in Fig. 4.26.

4.4 Wang and Mendel (1991) Exercise 3.16 described the truck backing up problem and it provided a set of 27 IF-THEN rules in a relational matrix. Those rules were obtained by applying the WM method to 14 different trajectories, each starting at different initial values for the two states, angular position, ϕ, and horizontal position, x. This exercise provides the data (in the

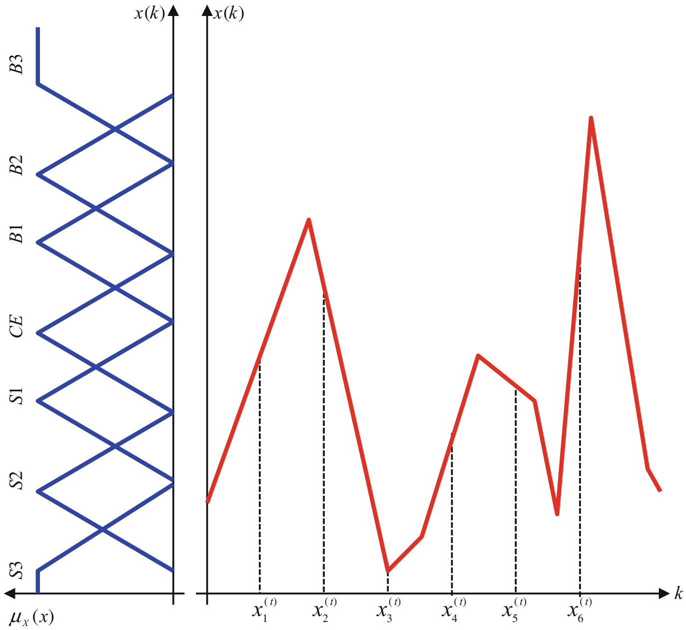


Fig. 4.26 Time series and MFs for Exercise 4.3

table below) for two of these trajectories, and asks you to *apply the WM method to create rules and a rule degree for each of them.*

To do this, use the following MF data (repeated here for the convenience of the reader): The MFs for ϕ are normal isosceles triangles with end points a, b, where: $(a_{S3} = -115°, b_{S3} = -15°)$, $(a_{S2} = -45°, b_{S2} = 45°)$, $(a_{S1} = 15°, b_{S1} = 90°)$, $(a_{CE} = 80°, b_{CE} = 100°)$, $(a_{B1} = 90°, b_{B1} = 165°)$, $(a_{B2} = 135°, b_{B2} = 225°)$, and $(a_{B3} = 195°, b_{B3} = 295°)$. The MFs for $S1$, CE and $B1$ of x are also normal isosceles triangles, where: $(a_{S1} = 4, b_{S1} = 10)$, $(a_{CE} = 9, b_{CE} = 11)$ and $(a_{B1} = 10, b_{B1} = 16)$; $S2$ and $B2$ are left and right normal trapezoidal shoulder MFs, respectively, where (see Fig. 2.21): $S2 = (0, 0, 1.5, 7)$ and $B2 = (13, 18.5, 20, 20)$. The MFs for $S2$, $S1$, CE, $B1$, and $B2$ of θ are also normal isosceles triangles, where: $(a_{S2} = -33°, b_{S2} = -7°)$, $(a_{S1} = -14°, b_{S1} = 0°)$, $(a_{CE} = -4°, b_{CE} = 4°)$, $(a_{B1} = 0°, b_{B1} = 14°)$, and $(a_{B2} = 7°, b_{B2} = 33°)$; $S3$ and $B3$ are left and right normal trapezoidal (triangle) shoulder MFs, respectively, where $S3 = (-40°, -40°, -40°, -20°)$ and $B3 = (20°, 40°, 40°, 40°)$.

(a) $(x_0, \phi_0) = (1, 0°)$
(b) $(x_0, \phi_0) = (1, 90°)$
(c) Combine the two sets of rules obtained in parts (a) and (b).

Data for Exercise 4.4(a)				Data for Exercise 4.4(b)			
t	x	ϕ°	θ°	t	x	ϕ°	θ°
0	1.00	0.00	−19.00	0	1.00	90.00	18.00
1	1.95	9.37	−17.95	1	1.15	81.11	16.00
2	2.88	18.23	−16.90	2	1.43	73.19	14.00
3	3.79	26.59	−15.85	3	1.83	66.24	12.00
4	4.65	34.44	−14.80	4	2.31	60.27	10.00
5	5.45	41.78	−13.75	5	2.88	55.29	8.00
6	6.18	48.60	−12.70	6	3.50	51.30	6.00
7	7.48	54.91	−11.65	7	4.16	48.31	4.00
8	7.99	60.71	−10.60	8	4.86	46.31	2.00
9	8.72	65.99	−9.55	9	5.56	45.31	0.00
10	9.01	70.75	−8.50	10	6.26	45.31	−2.00
11	9.28	74.98	−7.45	11	6.95	46.31	−4.00
12	9.46	78.70	−6.40	12	7.61	48.31	−6.00
13	9.59	81.90	−5.34	13	8.23	51.30	−8.00
14	9.72	84.57	−4.30	14	8.79	55.29	−10.00
15	9.81	86.72	−3.25	15	9.28	60.27	−12.00
16	9.88	88.34	−2.20	16	9.67	66.24	−14.00
17	9.91	89.44	0.00	17	9.95	73.19	−16.00
				18	10.09	81.11	−18.00
				19	10.09	90.00	0.00

4.5 Derive the normal equations in (4.17).

4.6 Define a squared error function that depends on all N training data. Redevelop (4.30) and (4.35) for this new error function.

4.7 Beginning with $e^{(t)}$ in (4.26) but for the singleton TSK fuzzy system in Example 4.6, derive the steepest descent algorithms for all of its antecedent and consequent parameters.

4.8 In Example 4.7, provide the steepest-descent algorithm for $\sigma_{F_i^l}$.

4.9 Repeat Example 4.7 for non-singleton fuzzification and the following two cases: (1) σ_{X_i} are different for $i = 1, \ldots, p$, and (2) $\sigma_{X_i} = \sigma_X$ for $i = 1, \ldots, p$ [Hint for (2): Use the chain rule very carefully, because the same σ_X appears in all of the FBFs.].

4.10 For the TSK fuzzy system that is in Example 4.6, develop an SVD-QR design based on $\mathbf{G}$ in (4.22). Interpret this design in terms of (4.19) and the $\mathbf{g}(\mathbf{x}^{(t)})$ in (4.21), i.e., explain what is being reduced by it.

4.11 Repeat Example 4.8 for a Mamdani particle when $\sigma_{X_i} = \sigma_X (i = 1, \ldots, p)$.

4.12 Repeat Example 4.8 for a TSK particle, when $y_{\mathrm{TSK}}(\mathbf{x})$ is given in (4.3) and (4.4).

4.13 In Sect. 4.3, the way in which the training data has been established to forecast a time-series is to ensure maximum overlap between successive training elements. Many other ways can be created to use the training data that either do not have so much overlap, or do not have any overlap at all between successive training elements.

(a) For the same N data points, create a training set that advances two points to the right, from one element in the training set to the next, instead of just one point to the right [as in (4.48)]. Suppose that the training set is to consist of 50% of all the data. What are the rules for this set of training data? How many rules will there be? What are the testing elements?

(b) For the same N data points, create a training set that has no overlap from one element in the training set to the next. As in part (a), suppose that the training set is to consist of 50% of all the data. What are the rules for this set of training data? How many rules will there be? What are the testing elements?

4.14 Explain how a FLA can be used to make an engineering judgment decision or action, e.g. as in global warming.

4.15 Establish all of the numerical values that are needed to completely specify the five MFs depicted in Fig. 4.8.

4.16 Using the MFs in Fig. 4.8 for x_1 and x_2 and focusing on two-antecedent rules:

(a) Draw the type-1 first-order rule partition diagram (see Example 3.3). How many type-1 first-order rule partitions are there?

(b) Draw the type-1 second-order rule partition diagram (see Example 3.3). How many type-1 second-order rule partitions are there?

4.17 Explain how to go from the data in Table 4.10 to Gaussian MFs for all five labels.

4.18 Examine Fig. 4.8 and explain why it is possible for a single-antecedent FLA to fire 1, 2, or 3 rules, and for a two-antecedent FLA to fire 1, 2, 3, 4, 6, or 9 rules.

4.19 As in Example 4.9, compute $y_{c1}(2, 4)$.

4.20 Flowchart the formulas needed to optimize the type-1 fuzzy RBC described in Sect. 4.6.4.

4.21 Obtain a formula for the Mamdani FPID controller in Table 4.14, using singleton fuzzification, product implication and COS defuzzification.

References

Adas, A.M. 1998. Using adaptive linear prediction to support real-time VBR video under RCBR network service model. *IEEE Transactions on Networking* 6: 635–644.

Arabshahi, P., J.J. Choi, R.J. Marks, II, and T.P. Caudell. 1996. Fuzzy parameter adaptation in optimization: Some neural net training examples. *IEEE Computational Science & Engineering* 57–65 (Spring).

Astrom, K.J., and T. Hagglund. 2001. The future of PID control. *Control Engineering Practice* 9 (11): 1163–1175.

Astrom, K. J., and T. Hagglund. 2005. *Advanced PID control*. ISA.

Autonne, L. 1902. Sur Les Groupes Lineaires, Reels et Orthogonaux. *Bulletin de la société mathématique de France* 30: 121–133.

Casdagli, M. 1992. A dynamical systems approach to modeling input–output systems. In *Nonlinear modeling and forecasting*. Vol. XII of SFI studies in the sciences of complexity Process, 265–281. New York: Addison-Wesley.

Castillo, O., and P. Melin. 2012. Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review. *Information Sciences* 205: 1–19.

Castillo, O., R.M.-Marroquin, P. Melin, F. Valdez, and J. Soria. 2012. Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. *Information Sciences* 192: 19–38.

Chen, S., S.A. Billings, and W. Luo. 1989. Orthogonal least squares methods and their application to nonlinear system identification. *International Journal of Control* 50: 1873–1896.

Chen, S., C.F.N. Cowan, and P.M. Grant. 1991. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks* 2: 302–309.

Chiu, S. 1994. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems* 2: 267–278.

Chiu, S. 1997a. Extracting fuzzy rules from data for function approximation and pattern classification. In *Fuzzy information engineering: A guided tour of applications*, ed. D. Dubois, H. Prade and R. Yager, Ch. 9. New York: Wiley.

Chiu, S. 1997b. An efficient method for extracting fuzzy classification rules from high dimensional data. *Journal of Advanced Computational Intelligence* 1 (1): 1–7.

Chu, P., and J.M. Mendel. 1994. First break refraction event picking using fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 2: 255–266.

Cox, E.A. 1995. *Fuzzy logic for business and industry*. Rockland, MA: Charles River Media.

Duan, X.-G., H.-X. Li, and H. Deng. 2008. Effective tuning method for fuzzy PID with internal model control. *Industrial and Engineering Chemistry Research* 47: 8317–8323.

Duda, R.O. 1994. Elements of pattern recognition. In *A prelude to neural networks: Adaptive and learning systems*, ed. J.M. Mendel, 3–33. Englewood-Cliffs, NJ: Prentice-Hall.

Duda, R.O., P.E. Hart, and D.G. Stork. 2001. *Pattern classification*, 2nd ed. New York: Wiley.

Eckart, C., and G. Young. 1939. A principal axis transformation for non-hermitian matrices. *Bulletin of the American Mathematical Society* 45: 118–121.

Farmer, J.D. 1982. Chaotic attractors of infinite-dimensional dynamical systems. *Physica* 4-D: 366–393.

Feng, G. 2006. A survey an analysis and design of model-based fuzzy control systems. *IEEE Transactions on Fuzzy Systems* 14: 676–697.

Francis, B.A., and W.M. Wonham. 1976. The internal model principle of control theory. *Automatica* 12: 457–465.

Galichet, S., and L. Foulloy. 1995. Fuzzy controllers: Synthesis and equivalences. *IEEE Transactions on Fuzzy Systems* 3: 140–148.

Golub, G.H., and C.F. Van Loan. 1983. *Matrix computations*. Baltimore, MD: Johns Hopkins Univ. Press.

Haykin, S. 1996. *Adaptive filter theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall.

Hirota, K. 1995. History of industrial applications of fuzzy logic in Japan. In *Industrial applications of fuzzy logic and intelligent systems*, ed. J. Yen, R. Langari, L.A. Zadeh, 43–54. IEEE Press.

Hohensohn, J., and J.M. Mendel. 1994. Two-pass orthogonal least-squares algorithm to train and reduce fuzzy logic systems. In *Proceedings of the third IEEE conference on fuzzy systems*, vol. 1, 696–700, Orlando, FL.

Hohensohn, J., and J.M. Mendel. 1996. Two-pass orthogonal least-squares algorithm to train and reduce the complexity of fuzzy logic systems. *Journal of Intelligent and Fuzzy Systems* 4: 295–308.

Holmblad, L., and I. Ostergaard. 1982. Control of a cement kiln by fuzzy logic. In *Fuzzy information and decision-processes*, ed. M.M. Gupta and E. Sanchez, 389–399. Amsterdam, The Netherlands: North-Holland.

Horikawa, S., T. Furahashi, and Y. Uchikawa. 1992. On fuzzy modeling using fuzzy neural networks with back-propagation algorithm. *IEEE Transactions on Neural Networks* 3: 801–806.

Hu, B., G.K.I. Mann, and R.G. Gasine. 2001. A systematic study of fuzzy PID controllers—Function-based evaluation approach. *IEEE Transactions on Fuzzy Systems* 9 (5): 699–711.

Huang, T.T., H.Y. Chung, and J.J. Lin. 1999. A fuzzy PID controller being like parameter varying PID. In *Proceedings of FUZZ-IEEE 1999*, 269–275, Seoul, Korea.

Jang, J.-S.R. 1992. Self-learning fuzzy controllers based on temporal back-propagation. *IEEE Transactions on Neural Networks* 3: 714–723.

Jang, J.-S.R. 1993. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics* 23: 665–684.

Jang, J.-S.R., and C.-T. Sun. 1995. Neuro-fuzzy modeling and control. *Proceedings of the IEEE* 83: 378–406.

Jang, J.-S.R., C.-T. Sun, and E. Mizutani. 1997. *Neuro-fuzzy and soft-computing*. Upper Saddle River, NJ: Prentice-Hall.

Karnik, N.N., and J.M. Mendel. 1998. *An introduction to type-2 fuzzy logic systems*. USC-SIPI Report #418, Univ. of Southern Calif., Los Angeles, CA, June 1998. This can be accessed at: http://sipi.usc.edu/research; then choose “sipi technical reports/418”.

Kennedy, J., and R. Eberhart. 1995. Particle swarm optimization. In *Proceedings of IEEE international conference on neural network*, 1942–1948.

Klema, V.C., and A.J. Laub. 1980. The singular-value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control* AC-25: 164–176.

Krunz, M., R. Sass, and H. Hughes. 1995. Statistical characteristics and multiplexing of mpeg streams. In *Proceedings of IEEE international conference on computer communications, INFOCOM’95*, vol. 2, Boston, MA, 455–462.

Kumbasar, T., and H. Hagras. 2015. Interval type-2 fuzzy PID controllers. In *Springer handbook of computational intelligence*, ed. J. Kacprzyk and W. Pedrycz, Chapter 18, 285–294. New York: Springer.

Kuncheva, L.I. 2000. *Fuzzy classifier design*. Heidelberg: Physica-Verlag.

Lapedes, A.S., and R. Farber. 1987. *Nonlinear signal processing using neural networks: Prediction and system modeling*. Technical report LA-UR-87–2662, Los Alamos National Lab., Los Alamos, NM.

Lee, C.-C. 1990. Fuzzy logic in control systems: Fuzzy logic controller, Part II. *IEEE Transactions on Systems, Man, and Cybernetics*. SMC-20: 419–435.

Li H.X., and H.B. Gatland. 1996. Conventional fuzzy control and its enhancement. *IEEE Transactions on Systems, Man, and Cybernetics—Part B* 26(5): 791–797.

Liang, Q., and J.M. Mendel. 2001. MPEG VBR video traffic modeling and classification using fuzzy techniques. *IEEE Transactions on Fuzzy Systems* 9: 183–193.

Lin, C.-T., and C.S.G. Lee. 1996. *Neural fuzzy systems: A neuro-fuzzy synergism to intelligent systems*. Upper Saddle River, NJ: Prentice-Hall PTR.

MacDuffee, C.C. 1933. *The theory of matrices*. New York: Springer.

Mackey, M.C., and L. Glass. 1977. Oscillation and chaos in physiological control systems. *Science* 197: 287–289.

Mamdani, E.H. 1994. Fuzzy control—A misconception of theory and application. *IEEE Expert-A Fuzzy Logic Symposium* 9 (4): 27–28.

Mamdani, E.H., and S. Assilian. 1975. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* 7: 1–13.

Manzoni, P., P. Cremonesi, and G. Serazzi. 1999. Workload models of VBR video traffic and their use in resource allocation policies. *IEEE Transactions on Networking* 7: 387–397.

Mendel, J.M. 1994. *A prelude to neural networks: Adaptive and learning systems*. Englewood-Cliffs, NJ: Prentice-Hall.

Mendel, J.M. 1995a. Fuzzy logic systems for engineering: A tutorial. *Proceedings of the IEEE* 83: 345–377.

Mendel, J.M. 1995b. *Lessons in estimation theory for signal processing, communications and control*. Englewood Cliffs, NJ: Prentice-Hall PTR.

Mendel, J.M. 2001. *Introduction to rule-based fuzzy logic systems*. Upper Saddle River, NJ: Prentice-Hall.

Mendel, J.M. 2002. An architecture for making judgments using computing with words. *International Journal of Applied Mathematics and Computer Science* 12 (3): 325–335.

Mendel, J.M. 2007. Computing with words: Zadeh, turing, popper and occam. *IEEE Computational Intelligence Magazine* 2: 10–17.

Mendel, J.M. 2014. General type-2 fuzzy logic systems made simple: A tutorial. *IEEE Transactions on Fuzzy Systems* 22: 1162–1182.

Mendel, J.M., and K.S. Fu, ed. 1970. *Adaptive, learning and pattern recognition systems: Theory and applications*. Academic Press, Inc.

Mendel, J.M., and G.C. Mouzouris. 1997. Designing fuzzy logic systems. *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing* 44: 885–895.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: Aiding people in making subjective judgments*. Hoboken, NJ: Wiley and IEEE Press.

Mendel, J.M., S. Murphy, L.C. Miller, M. Martin, and N. Karnik. 1999. The fuzzy logic advisor for social judgments. In *Computing with words in information/intelligent systems*, ed. L.A. Zadeh and J. Kacprzyk, 459–483. Physica-Verlag.

Mendel, J.M., H. Hagras, W.-W. Tan, W.W. Melek, and H. Ying. 2014. *Introduction to type-2 fuzzy logic control*. Hoboken, NJ: John Wiley and IEEE Press.

Moody, J. 1989. Fast learning in multi-resolution hierarchies. In *Advances in neural information processing systems I*, ed. D.S. Touretzky, Chapter 1, 29–39. San Mateo, CA: Morgan Kaufman.

Moody, J., and C.J. Darken. 1989. Fast learning in networks of locally-tuned processing units. *Neural Computation* 1: 281–294.

Moon, B.S. 1995. Equivalence between fuzzy logic controllers and PI controllers for single input systems. *Fuzzy Sets and Systems* 69: 105–113.

Mouzouris, G.C., and J.M. Mendel. 1996. Designing fuzzy logic systems for uncertain environments using a singular-value–QR decomposition method. In *Proceedings of the fifth IEEE international conference on fuzzy systems*, New Orleans, LA.

Mouzouris, G.C., and J.M. Mendel. 1997. A singular-value–QR decomposition based method for training fuzzy logic systems in uncertain environments. *Journal of Intelligent and Fuzzy Systems* 5: 367–374.

Munakata, T., and Y. Jani. 1994. Fuzzy systems: An overview. *Communications of the ACM* 37: 69–96.

Palm, R. 1992. Sliding mode fuzzy control. In *Proceedings of 1992 IEEE international conference on fuzzy systems*, 519–526, San Diego, CA.

Quinney, D. 1985. *An introduction to the numerical solution of differential equations*. England: Research Studies Press.

Qiao, W.Z., and M. Mizumoto. 1996. PID type fuzzy controller and parameters adaptive method. *Fuzzy Sets and Systems* 78: 23–35.

Rasband, S.N. 1990. *Chaotic dynamics of non-linear systems*. New York: Wiley.

Rickard, J.T., J. Aisbett, R. Yager, and G. Gibbon. 2011. Linguistic weighted power means: Comparison with the linguistic weighted average. In *Proceedings of FUZZ-IEEE 2011*, 2011 World Congress on Computational Intelligence, 2185–2192, Taipei, Taiwan.

Rickard, J.T., J. Aisbett, R.R. Yager, and G. Gibbon. 2013. Computing with words using weighted power mean aggregation operators. In *Soft computing: State of the art theory and novel applications*, ed. R.R. Yager, A.M. Abbasov, M.Z. Reformat, and S. Shahbazova, 145–160. New York: Springer.

Rose, O. 1995. *Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems*. Univ. of Wurzburg, Institute of Computer Science, Research Report 101.

Rutkowski, L. 2004. *Flexible neuro-fuzzy systems: Structures, learning and performance evaluation*. Boston: Kluwer.

Sanger, T.D. 1991. A tree-structured adaptive network for function approximation in high-dimensional spaces. *IEEE Transaction on Neural Networks* 2: 285–293.

Seborg, D.E., F.E. Thomas, and A.M. Duncan. 2004. *Process dynamics and control*, 2nd ed. New York: Wiley.

Setnes, M., and H. Hellendoorn. 2000. Orthogonal transforms for ordering and reduction of fuzzy rules. In *Proceedings of FUZZ-IEEE'00*, 700–705, San Antonio, TX.

Simon, D. 2013. *Evolutionary optimization algorithms*. Hoboken, NJ: Wiley.

Skogestad, S. 2003. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 13 (4): 291–309.

Stewart, G.W. 1973. *Introduction to matrix computations*. New York: Academic Press.

Sugeno, M. 1985. An introductory survey of fuzzy control. *Information Sciences* 36: 59–83.

Sugeno, M., and T. Yasukawa. 1993. A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* 1: 7–31.

Sun, J., X. Wu, V. Palade, W. Fang, C.H. Lai, and W. Xu. 2012. Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Information Sciences* 193: 81–103.

Vaccaro, R. (ed.). 1991. *SVD and signal processing algorithms, II, algorithms, analysis and applications*. New York: Elsevier.

Wang, L.-X. 1992. *Analysis and design of fuzzy systems*. Ph.D. dissertation, University of Southern California, Los Angeles, CA.

Wang, L.-X. 1994. *Adaptive fuzzy systems and control: Design and stability analysis*. Englewood Cliffs, NJ: PTR Prentice-Hall.

Wang, L.-X. 2003. The WM method completed: A flexible fuzzy system approach to data mining. *IEEE Transactions on Fuzzy Systems* 11 (6): 768–782.

Wang, L.-X., and J.M. Mendel. 1991. *Generating fuzzy rules from numerical data, with applications*. USC-SIPI Report #169, January 1991.

Wang, L.-X., and J.M. Mendel. 1992a. Fuzzy basis functions, universal approximation, and orthogonal least squares learning. *IEEE Transactions on Neural Networks* 3: 807–813.

Wang, L.-X., and J.M. Mendel. 1992b. Back-propagation of fuzzy systems as non-linear dynamic system identifiers. In *Proceedings of IEEE international conference on fuzzy systems*, 1409–1418, San Diego, CA.

Wang, L.-X., and J.M. Mendel. 1992c. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics* 22: 1414–1427.

Wang, X., Y. He, L. Dong, and H. Zhao. 2011. Particle swarm optimization for determining fuzzy measures from data. *Information Sciences* 181: 4230–4252.

Wei, F., S. Jun, X.Z.-Ping, and W.-B. Xu. 2010. Convergence analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter. *Acta Physica Sinica* 59(6): 3686–3694.

Wu, D., and J.M. Mendel. 2007. Aggregation using the linguistic weighted average and interval type-2 fuzzy sets. *IEEE Transactions on Fuzzy Systems* 15 (6): 1145–1161.

Yager, R., and D. Filev. 1994. Generation of fuzzy rules by mountain clustering. *Journal of Intelligent and Fuzzy Systems* 2: 209–219.

Yasunobu, S., and S. Miyamoto. 1985. Automatic train operation system by predictive fuzzy control. In *Industrial applications of fuzzy control*, ed. M. Sugeno. North-Holland, Amsterdam, The Netherlands: Elsevier Science.

Yen, J., and R. Langari. 1999. *Fuzzy logic: Intelligence, control, and information*. Upper Saddle River, NJ: Prentice-Hall.

Yen, J., and L. Wang. 1996. An SVD-based fuzzy model reduction strategy. In *Proceedings of the fifth international conference on fuzzy systems*, 835–841, New Orleans, LA.

Yen, J., and L. Wang. 1999. Simplifying fuzzy rule-based models using orthogonal transformations. *IEEE Transactions on Systems, Man and Cybernetics* 29, Part B.

Ying, H. 2000. *Fuzzy control and modeling: Analytical foundations and applications*. Piscataway, NJ: IEEE Press.

Zadeh, L.A. 1996. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems* 4: 103–111.

Chapter 5
Sources of Uncertainty

5.1 Uncertainties in a Fuzzy System

The title of this book, *Uncertain Rule-Based Fuzzy Systems*, emphasizes the word *uncertain* using it as its first word. Note, however, that this word applies to the rest of the title, *rule-based fuzzy systems*. Hence, this chapter does not provide a lot of general discussions about the meaning of uncertainty; instead, it only examines its meaning relative to (conditioned on) a rule-based fuzzy system.

5.1.1 Uncertainty: General Discussions

Uncertainty comes in many guises and is independent of what kind of fuzzy logic, or any kind of methodology, one uses to handle it. One of the best sources for general discussions about uncertainty is Klir and Wierman (1998). The late Professor George Klir and his students, e.g., Klir and Folger (1988) focused on uncertainty topics for more than 30 years, since the 1980s. Klir and Wierman (1998) represents an amalgamation and sharpening of the many ideas from their works. Regarding the *occurrence of uncertainty*, they state (Klir and Wierman 1998, p. 2):

> When dealing with real-world problems, we can rarely avoid uncertainty. At the empirical level, uncertainty is an inseparable companion of almost any measurement, resulting from a combination of inevitable measurement errors and resolution limits of measuring instruments. At the cognitive level, it emerges from the vagueness and ambiguity inherent in natural language. At the social level, uncertainty has even strategic uses and it is often created and maintained by people for different purposes (privacy, secrecy, propriety).

Regarding the *causes of uncertainty*, they state Klir and Wierman (1998, p. 5):

> Uncertainty involved in any problem-solving situation is a result of some information deficiency. Information (pertaining to the model within which the situation is

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_5

> conceptualized) may be incomplete, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way. In general, these various information deficiencies may result in different types of uncertainty.

Regarding the *nature of uncertainty*, they state Klir and Wierman (1998, p.43):

> Three types of uncertainty are now recognized… fuzziness[1] (or vagueness), which results from the imprecise boundaries of fuzzy sets; non-specificity[2] (or imprecision), which is connected with sizes (cardinalities) of relevant sets of alternatives; and strife[3] (or discord), which expresses conflicts among the various sets of alternatives.

They divide these three types of uncertainty into two major classes, fuzziness and ambiguity, where ambiguity ("one to many relationships") includes non-specificity and strife.

Another source for some general discussions about uncertainty is Berenji (1988, p. 233), who states, in agreement with Klir and Wierman (1998), that "uncertainty stems from lack of complete information." He also states "Uncertainty may also reflect incompleteness, imprecision, missing information, or randomness in data and a process."

Taken out of context, "uncertainty" is relatively abstract because of its many varieties; however, as is demonstrated next, taken in the context of a fuzzy set or a fuzzy system, uncertainty is easy to understand.

5.1.2 Uncertainties and Sets

It seems paradoxical Mendel (2003) that the word fuzzy, which has the connotation of uncertainty, is associated with a set—a type-1 fuzzy set—whose MF is completely certain once its parameters are specified. The following quote from Klir and Folger (1988) uses the word paradoxical:

---

[1]Klir and Wierman (1998, p. 103) associate the following synonyms with fuzziness: "vagueness, cloudiness, haziness, unclearness, indistinctness and sharplessness."

[2]Klir and Wierman (1998, p. 103) associate the following synonyms with non-specificity: "variety, generality, diversity, equivocation and imprecision." The use of *imprecision* both as a synonym for non-specificity and in the discussion about fuzziness is confusing. In a private correspondence to me, Klir states "When it comes to the term 'imprecision,' it is meaningful to use it (and it is often used) for both non-specificity and fuzziness. When it is used for non-specificity, it refers to information—based imprecision; here, uncertainty results from information deficiency. When it is used for fuzziness, it refers to linguistic imprecision. Clearly, the term 'imprecision' has been used prior to the emergence of fuzzy sets. For example, we talk about limited precision (or imprecision) of measurements (or computations)…. Non-specificity and imprecision are thus connected, but… we should not consider them as synonyms. The reason is that imprecision is also connected with fuzziness. This ambiguity can be avoided by distinguishing information-based imprecision (equivalent to non-specificity) with linguistic imprecision (equivalent to fuzziness)."

[3]Klir and Wierman (1998, p. 103) associate the following synonyms with strife: "dissonance, incongruency, discrepancy, conflict, and discord."

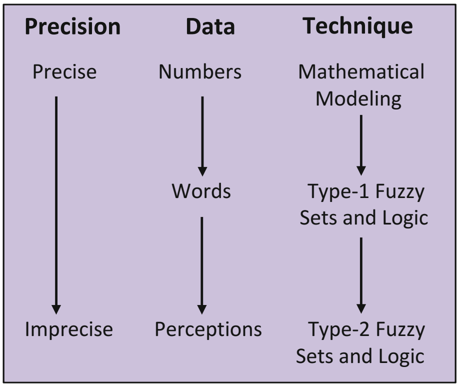


Fig. 5.1 Relationships between imprecision, data, and fuzzy techniques (adapted from John and Coupland (2007) ©, 2007 IEEE)

> The accuracy of any MF is necessarily limited. In addition, it may seem problematical, if not *paradoxical*, that a representation of fuzziness is made using membership grades that are themselves precise real numbers. Although this does not pose a serious problem for many applications, it is nevertheless possible to extend the concept of a fuzzy set to allow for the distinction between grades of membership to become blurred.

Type-2 fuzzy sets resolve this paradox because they formalize the blurring of MF grades. This will be made very clear beginning with Chap. 6.

John and Coupland (2007) have a very interesting figure (Fig. 5.1) showing the relationships between precision, data, and technique. Problems can range from precise to imprecise; data can range from numbers to words to perceptions, and techniques can range from mathematical modeling to type-1 fuzzy sets and logic to type-2 fuzzy sets and logic. According to John and Coupland (2007),

> The more imprecise or vague the data, then type-2 fuzzy sets should offer a significant improvement on type-1 fuzzy sets. … As the level of imprecision increases, then type-2 fuzzy logic [sets] provide a powerful paradigm for tackling the problem. Problems that contain crisp, precise data do not, in reality, exist. However, some problems can be tackled effectively using mathematical techniques where the assumption is that the data are precise. Other problems … use imprecise terminology that can often be effectively modeled using type-1 fuzzy sets. Perceptions … are at a higher level of imprecision and type-2 fuzzy sets [and logic] can effectively model this imprecision.

The rest of this book subscribes to their concluding sentence, but also advocates using type-2 fuzzy sets in situations where there is uncertainty about numbers (e.g., due to measurement errors or noise).

5.1.3 Uncertainties in a Fuzzy System

The following sources of uncertainty can occur for the rule-based fuzzy system in Fig. 1.2:

- Uncertainty about the meanings of the words that are used in the rules
- Uncertainty about the consequent that is used in a rule
- Uncertainty about the measurements that activate the fuzzy system
- Uncertainty about the data that are used to tune the parameters of a fuzzy system.

It was already mentioned in footnote 17 of Chap. 4 that Zadeh (1996, 1999) has advocated "computing with words" (CWW) and using fuzzy sets to do this. In addition to that footnote, Zadeh (1996, p. 103) states:

- Computing with words is a necessity when the available information is too imprecise to justify the use of numbers and… when there is a tolerance for imprecision which can be exploited to achieve tractability, robustness, low solution cost, and better rapport with reality.
- Fuzzy logic is a methodology for computing with words.
- As used by humans, words have fuzzy denotations.
- A key aspect of CWW is that it involves a fusion of natural languages and computation with fuzzy variables.

In a December 26, 2008 e-mail, Zadeh further stated:

> In 2008, Computing with Words … has grown in visibility and recognition. There are two basic rationales for the use of Computing with Words. First, when we have to use words because we do not know the numbers. And second, when we know the numbers but the use of words is simpler and cheaper, or when we use words to summarize numbers. In large measure, the importance of Computing with Words derives from the fact that much of human knowledge is described in natural language. In one way or another, the fuzzy-logic-based machinery of Computing with Words opens the door to a wide-ranging enlargement of the role of natural languages in scientific theories, including scientific theories which relate to economics, medicine, law and decision analysis.

Our thesis is that[4] *words mean different things to different people* and so there is *uncertainty* associated with words, which means that a fuzzy system must somehow use this uncertainty when it computes with words Mendel (1999). As mentioned in Sect. 5.1.2, type-1 fuzzy sets handle uncertainties about the meanings of words using *precise* MFs that the user believes capture the uncertainty of the words. Once the type-1 MFs have been chosen, all uncertainty about the words disappears, because type-1 MFs are totally precise. Type-2 fuzzy sets, on the other hand, handle[5] uncertainties about the meanings of words by *modeling* the uncertainties. This is accomplished, as is described in great detail in Chap. 6, e.g., by blurring the boundaries of type-1 MFs into what is called a *footprint of uncertainty* (see, also, Definition 1.4). Although a type-2 MF will also be totally precise, it includes the footprint of uncertainty that provides new degrees of freedom that let uncertainties be handled by a type-2 fuzzy system in totally new ways.

---

[4]Exactly what is meant by "Words mean different things to different people" is discussed in Sect. 5.2 at great length.

[5]By "handle" is meant to *model and minimize the effect of* (Mendel 2003).

In rules, one distinguishes between antecedent and consequent words, but in a rule there are also connector words (e.g., *and*, *or*). As stated in Wu and Mendel (2004):

> The uncertainties contained in linguistic connector words are due to the subtleness of people's thinking and the vagueness of natural language, as reflected by the fact that the linguistic connector word *and* (*or*) cannot be precisely modeled by a single t-norm (t-conorm) under all circumstances, and sometimes it even does not correspond to any t-norm or t-conorm, but some operation between them (Zimmerman and Zysno 1980).

In this book, no uncertainty is assigned to such connector words, because to do so [e.g., as in Zimmerman and Zysno (1980), Türksen (1986), (1992), Wu and Mendel (2004)] complicates a fuzzy system enormously.

The uncertainty about the meanings of the words that are used in rules seems to be in accord with *fuzziness*, which results from imprecise boundaries of fuzzy sets.

Consequents for rules are either obtained from experts, by means of knowledge mining (engineering), or are extracted directly from data. Because experts do not all agree, a survey of experts will usually lead to a histogram of possibilities for the consequent of a rule. This histogram represents the uncertainty about the consequent of a rule, and this kind of uncertainty is different from that associated with the meanings of the words used in the rules. A histogram of consequent possibilities can be handled by a type-2 fuzzy system.

Uncertainty about the consequent used in a rule, as established by a histogram of possibilities, seems to be in accord with *strife*, which expresses conflicts among the various sets of alternatives.

Measurements are usually corrupted by noise; hence, they are uncertain. The traditional ideas about noisy measurements, i.e., measurement = signal + noise, are not abandoned in this book. What is abandoned is the frequently made assumption of a priori knowledge of a probability model (i.e., a probability density function) for either the signal or the noise. Doing this gets around the major shortcoming of a probability-based model, namely the assumed probability model, for which results will be good if the data agree with the model, but may not be so good if the data do not. Uncertain measurements can be handled very naturally within the framework of a fuzzy system.[6]

Uncertain measurements (i.e., randomness in the data) can be modeled as fuzzy sets (type-1 or type-2); hence, uncertainty about the measurements that activate the fuzzy system seems to be in accord with *non-specificity* when non-specificity is associated with information-based imprecision, as suggested by Klir in this chapter's footnote 2.

---

[6]This author has spent much of his career working with probability-based models [e.g., Mendel (1983), (1990), (1995)] and is in no way recommending that such models be abandoned. When one has a high confidence in the validity of probability-based models, they can provide excellent results. Type-2 fuzzy systems provide a viable alternative to them when one does not have a high confidence in the validity of a probability-based model or cannot determine such a model because of system complexities, such as non-linearity, time-variability, or non-stationarity.

Finally, a fuzzy system contains many design parameters whose values must be set by the designer before the fuzzy system is operational. There are many ways to do this, and most make use of a training set of data, which consists of input–output pairs for the fuzzy system. If those pairs are measured signals, then they are as uncertain as the measurements that excite the fuzzy system. In this case—one that is quite common in practice, but has not received enough attention in the fuzzy system literature—the fuzzy system must be tuned using unreliable data, which is yet another form of uncertainty that can be handled by a type-2 fuzzy system.

The uncertainty about the data that are used to tune the parameters of a fuzzy system also seems to be in accord with *non-specificity* when non-specificity is associated with information-based imprecision.

Based on these discussions, it would appear that a type-2 fuzzy system is able to directly address all three types of uncertainty—*fuzziness*, *strife*, and *non-specificity*. Klir and Wierman (1998), however, appear to be dismissive of type-2 fuzzy sets, when they state (1998, p. 20):

> These more general fuzzy sets [*fuzzy sets of type-2*]… are not relevant to our discussion of uncertainty measures and principles.

On this point we strongly disagree,[7] and, in fact, in this book, all of these kinds of uncertainty are examined, and it is shown how a type-2 fuzzy system—one that makes use of type-2 fuzzy sets—can be designed to perform well in spite of them. Type-2 fuzzy sets are the means to handling all three types of uncertainty totally within the framework of fuzzy set theory. They are able to do this because they directly model uncertainties.

5.2 Words Mean Different Things to Different People

Today, computing with words must still be done using numbers, and, therefore, numeric intervals must be associated with words (linguistic terms). Suppose, for example, that it has been established ahead of time that the words to be used in a collection of rules will all lie on the scale 0–10. An interesting question then is: *How can one cover the scale* 0–10 *with words?* In typical engineering applications of fuzzy sets, this question is not asked, because one chooses the number of fuzzy sets that will cover an interval arbitrarily, and then choose the names for these sets just as arbitrarily (e.g., *zero*, *small positive*, *medium positive*, and *large positive*). This works fine for many engineering applications, when rules are extracted from data. However, it is a questionable practice when rules are extracted from people (experts). Put another way, machines do not care about words, but people do.

[7]In a private correspondence to me, Klir clarified the intended meaning of this statement, in that his book is restricted only to type-1 fuzzy sets and, hence, the more general fuzzy sets are not relevant to the discussions in the book. The quoted statement was not intended to dismiss the significance of type-2 fuzzy sets.

In the rest of this section, in order to make our discussions concrete, the focus is on applying fuzzy sets to situations where the meanings of words are associated with a scale of 0–10. These discussions can easily be extended to arbitrary scales. Consider the following question, whose meaning will become clear in the following discussions:

> What is the *smallest* number of words (or phrases) that cover the interval 0–10?

It is demonstrated below that the answer to this question depends on whether or not uncertainty is associated with the words. Note that this question seems to be in accord with another aspect of *non-specificity* namely with the sizes of relevant sets of alternatives (cardinalities).

To answer this question, a survey was performed. With the help of a social scientist,[8] who is also an expert in survey methodology, 16 words were established that were thought to cover the interval 0–10. As is usually done for fuzzy sets and systems (Zadeh 1975), it was assumed that these words are generated in a context-free grammar.[9] The 16 words [which are referred to as *labels* (i.e., *terms*) in the survey] were randomized. Engineering undergraduate students were given the survey, whose wording was as follows[10]:

> Below are a number of labels that describe an interval or a "range" that falls somewhere between 0–10. For each label, please tell us where this range would start and where it would stop. (In other words, please tell us how much of the distance from 0 to 10 this range would cover.) For example, the range "quite a bit" might start at 6 and end at 8. It is important to note that not all the ranges are the same size.

Table 5.1 was provided to the students so that they only had to fill in two numbers for each label. The labels were randomized so that interval information collected about each label would be uncorrelated. A total of 87 surveys were completed; 17 had outliers (e.g., some students filled in 0–10 for the range of all the labels). Of the remaining 70, 40 were from men, 11 from women, and 19 were from

---

[8]Prof. Sheila Murphy, University of Southern California Annenberg School of Communications.

[9]Klir and Wierman (1998, p. 13) state "fuzzy numbers, fuzzy intervals, and other types of fuzzy sets give us enough flexibility to represent, as closely as desirable, states characterized by *expressions in a natural language* that are inherently vague. These expressions are, of course, strongly dependent on the context in which they are used. This implies that membership grade functions by which we attempt to capture the relevant linguistic expressions must be constructed in the context of each application." This raises the important question (to which no answer is provided), "Is there such a thing in the real world as a context-free grammar?".

[10]If a method that is used to collect word data from a group of subjects introduces uncertainties because the subjects do not understand something about the method (e.g., they do not understand what a MF is), then the method's uncertainties become comingled with the subject's uncertainties about the word(s) and this is not good, because the two kinds of uncertainties cannot be unraveled since no measure of the methodological uncertainties about a word is available. There is a modest size literature on how to elicit type-1 MFs from either a single subject or a group of subjects. See Appendix 3A in Mendel and Wu (2010) for a history and comparison of these methods. Asking subjects to provide interval endpoints for words, as is done in this book, does not introduce methodological uncertainties; it is easy for a subject to provide such data.

Table 5.1 Survey table with randomized labels

Range label	Start (a)	End (b)
None		
Some		
A good amount		
An extreme amount		
A substantial amount		
A maximum amount		
A fair amount		
A moderate amount		
A large amount		
A small amount		
Very little		
A lot		
A sizeable amount		
A bit		
A considerable amount		
A little bit		

students who chose not to identify their sex. The following results do not distinguish between the sex of the respondent, although clearly one could if that was felt to be important.[11] Survey results are summarized in Table 5.2.

Because a range was requested for each label, and a range is defined by two numbers—*start* (a) and *end* (b)—the survey led to sample statistics for these two numbers, namely their sample means (m_a and m_b) and sample standard deviations (σ_a and σ_b). The sample standard deviations represent the uncertainties associated with each label. Observe, in Table 5.2, that standard deviations are not the same for the *start* and *end* values for each label.

Uncertainty about a word is of two kinds [e.g., Mendel (2007a, b), Mendel and Wu (2010)]: (1) *intra-uncertainty*, which is the uncertainty that a person has about the word; and (2) *inter-uncertainty*, which is the uncertainty that a group of people have about the word. When only group statistics are used to map the sample mean and standard deviation of interval endpoints from a group of subjects into a type-1 fuzzy set, then the resulting type-1 fuzzy set model is only capturing the inter-uncertainty about the word (this is what was done in Sect. 4.4.3). How to obtain type-2 fuzzy set word models that capture both the inter- and intra-uncertainties of a word from this kind of data is discussed in Sect. 10.4.1.

Table 5.2 data are also summarized in Fig. 5.2. For each label there are two circles, located at m_a and m_b, with a solid line between them. The dashed lines to the left of the left-hand circles and to the right of the right-hand circles each terminate in a vertical bar equal to one sample standard deviation (σ_a or σ_b, respectively), listed in Table 5.2.

[11]This would suggest that gender differences should be considered when computing with words.

Table 5.2 Processed survey results: ordered labels

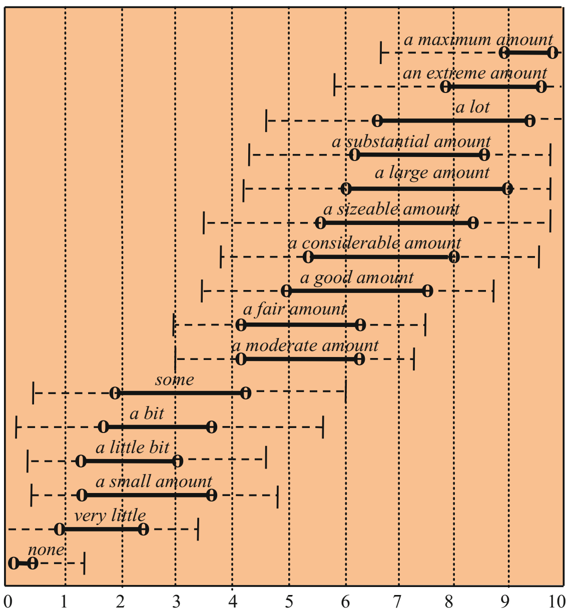
No.	Range label	Sample mean		Sample standard deviation	
		m_a	m_b	σ_a	σ_b
1	*None*	0.01	0.23	0.12	0.90
2	*Very little*	0.87	2.26	0.93	1.37
3	*A small amount*	1.30	3.64	0.87	1.18
4	*A little bit*	1.31	3.07	0.84	1.62
5	*A bit*	1.70	3.76	1.49	1.84
6	*Some*	1.83	4.29	1.24	1.67
7	*A moderate amount*	4.14	6.17	1.15	1.17
8	*A fair amount*	4.26	6.23	1.29	1.34
9	*A good amount*	4.94	7.44	1.49	1.37
10	*A considerable amount*	5.36	7.99	1.57	1.58
11	*A sizeable amount*	5.66	8.26	1.97	1.56
12	*A large amount*	6.09	8.90	1.66	1.04
13	*A substantial amount*	6.26	8.64	1.87	1.24
14	*A lot*	6.74	9.39	1.96	0.77
15	*An extreme amount*	7.89	9.60	2.10	0.77
16	*A maximum amount*	8.89	9.76	2.43	1.12

Observe, from Fig. 5.2, that

1. The dashed portions of the intervals for each label represent the label's uncertainty.
2. The 16 words cover the 0–10 interval, but only when label uncertainties are included.
3. There is a gap between the sample mean-value endpoints of *none* and *very little*, implying that either another word should be inserted between them or they should be combined. For illustrative purposes, in our following discussions, the latter is done.
4. People seem to agree that *none* starts at zero—and there is very little uncertainty about this (see σ_a for *none* in Table 5.2). To people, the word *none* seems to have a very strong connotation with the number "zero."
5. The same cannot be said for the label *a maximum amount*. The right-hand sample mean value for its range is 9.76 and not 10. One explanation for this is that people may be adverse to assigning the largest possible number to any label, because of an expectation that there could be another label that should have the largest number associated with it.[12] Because the labels were randomized, the students may have expected a phrase even stronger than *a maximum amount*, and they did not check the complete list of 16 words to see if such a stronger phrase actually occurred.

[12]Why, for example, if the top grade on an examination is *excellent* and the range for *excellent* is 8–10, are some people assigned an 8, and probably no one is assigned a 10? Perhaps, it is the expectation that someone else will do better, or that no one is perfect.

Fig. 5.2 All 16 labels and their intervals and uncertainty bands



6. There seems to be a linguistic gap between the labels *some* and *a moderate amount* as evidenced by the small degree of overlap between the sample mean endpoints of these labels. Perhaps this gap could be filled by including a new label called *somewhat moderate*.

What does one do with such uncertainty information when a type-1 fuzzy system is used to compute with words? Usually, a type-1 fuzzy set would explicitly *ignore* the uncertainty, either using just $[m_a, m_b]$ for a label, or, perhaps by being conservative, using the range associated with $[m_a - \sigma_a, m_b + \sigma_b]$. As Fig. 5.2 indicates, it is not correct to do the former, and, as explained next, it is also not correct to do the latter.

The dashed lines in Fig. 5.2 represent linguistic uncertainty in much the same way that standard deviation for a measured random quantity represents its uncertainty. When one works in the province of probability, it is useful and important to distinguish between the mean and the standard deviation; so, as has already been argued in Chap. 1, why should less be expected of us when one works in the province of fuzzy sets? Unfortunately, type-1 fuzzy sets cannot let us distinguish the dashed part of an interval from the solid part. Type-2 fuzzy sets can, and how to do this is explained in Sect. 10.4.

There is more that can be concluded from the survey results, namely,

1. In Fig. 5.3, the sets *none* and *very little* have been combined as explained earlier, and five labels are shown that cover the 0–10 range (except for the right-end anomaly), whose intervals between their sample mean start and

endpoints are drawn in very heavy. These are not the only five labels that could have been chosen. For example, instead of using *a moderate amount* one could have used *a fair amount*, or instead of using *a large amount* one could have used *a substantial amount*.

2. The intervals are not of equal size and there is more (or less) overlap between some than between others. MFs that are associated with these labels should make use of this information and be designed accordingly. How to do this is explained in Sect. 10.4.
3. It is possible to cover the 0–10 range with five labels, as indicated in Fig. 5.3; however, there is not much overlap between some of the label's sample mean endpoints. When the sample standard deviation information is used, then sufficient overlap is achieved.
4. It is also possible to cover the 0–10 interval with four or three labels (see Fig. 5.4). The smallest number of labels from the 16 labels used in our survey that cover the interval is three and this is only possible because of linguistic uncertainties; that is, overlap occurs for the three labels only because of uncertainty.[13]

Linguistic uncertainty appears to be useful in that it lets the 0–10 range be covered with a much smaller number of labels than without it. Put another way, in the context of firing rules in a fuzzy system, *uncertainty can fire rules*. This cannot occur in the framework of a type-1 fuzzy system, but it can occur in the framework of a type-2 fuzzy system. Additionally, *uncertainty can be used to control the rule explosion* that is so common in a fuzzy system. If, for example, one ignored uncertainty, and had rules with two antecedents, each of which has six labels, it would take 36 rules to completely describe the fuzzy rule base. On the other hand, using three labels for each antecedent requires a rule base with only nine rules. This is a 75% reduction in the size of the rule base. For rules with more than two antecedents the rule reduction is even greater.

This discussion concludes with a conjecture: *Uncertainty is good in that it lets people make decisions (albeit conservative ones) rapidly*. Perhaps this is why some people can make decisions very quickly and others cannot. The latter may have partitioned their variables into so many fine sets that they get hung up among the resulting enormous number of possibilities. They are the eternal procrastinators.

[13]In a private correspondence to the author, Klir suggests that two labels (see Fig. 5.2) *none to a small amount* and *large to a maximum amount* could also cover 0–10. Although this could be concluded by combining intervals (as was done for *none* and *very little*), it needs to be verified by means of another survey. It does seem quite plausible that a vocabulary that included these two terms would let the interval 0–10 be covered by as few as two labels. Klir also suggests that a single label such as *any amount* could cover 0–10, but then adds that this would have no pragmatic value (for a fuzzy system).

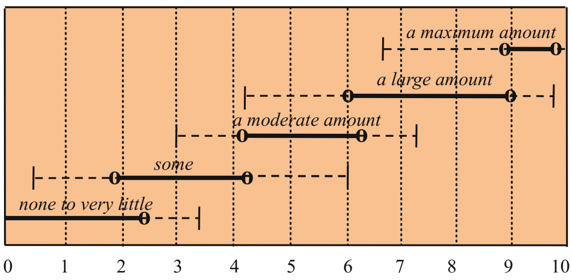


Fig. 5.3 Although five labels cover 0–10, there is not much overlap between some of them. It is when the sample standard deviation information is used that sufficient overlap is achieved

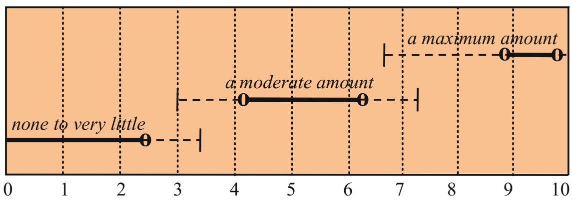


Fig. 5.4 The smallest number of labels that cover the interval 0-10 is three, and this is only possible because of uncertainties

This conjecture seems to be supported by Klir and Wierman (1998, page 4), when they state:

> Uncertainty has a pivotal role in any efforts to maximize the usefulness of systems models…. Uncertainty becomes very valuable when considered in connection to the other characteristics of systems models: a slight increase in uncertainty may often significantly reduce complexity and, at the same time, increase credibility of the model. Uncertainty is thus an important commodity in the modeling business, a commodity which can be traded for gains in the other essential characteristics of models.

Based on the results in Fig. 5.3, another survey was conducted using a different group of engineering students in which the five labels *none to very little*, *some*, *a moderate amount*, *a large amount*, and *a maximum amount* were randomized. The same question was asked as in our first survey, and results from 47 students are given in Table 5.3. Note that there is no longer an anomaly about reaching the maximum value of 10. This is probably because only five phrases were used and it was easy for the students to check them all to determine that the strongest one was *a maximum amount*.

It seems pretty clear from these surveys that *words do indeed mean different things to different people* and that by utilizing the uncertainties associated with words the number of alternatives can be reduced, which is in accord with the cardinality aspect of *non-specificity*.

Exercises

5.1 Take the survey in Table 5.1, and then compare your results with those given in Table 5.2. Explain how to modify the results given in Table 5.2 using your survey's results.

Table 5.3 Survey results for labels of fuzzy sets

No.	Range label	Sample mean		Sample standard deviation	
		m_a	m_b	σ_a	σ_b
1	*None to very little* (NVL)	0	1.99	0	0.81
2	*Some* (S)	2.54	5.25	0.91	1.37
3	*A moderate amount* (MOA)	3.64	6.46	0.88	0.86
4	*A large amount* (LA)	6.48	8.75	0.75	0.60
5	*A maximum amount* (MAA)	8.55	10	0.75	0

5.2 Choose a set of linguistic terms that may include some but not all of the linguistic terms used in Table 5.2 (i.e., include some new linguistic terms) and perform a survey, as was done to obtain the results in Table 5.2. Separate the results by gender and draw conclusions.

5.3 Create, administer, and analyze a survey that tests whether or not the results given in Tables 5.2 or 5.3 are context-independent.

5.4 If one does not have access to a group of subjects, or does not have access to a budget or the time to collect data from a group of subjects, or if the CWW application is for a *personal advisor* in which case data are collected only from that person, then data can be collected from a single subject by asking two similar questions like (Korjani and Mendel 2013, Mendel and Wu 2014): *Suppose that a word can be located on a scale of l to r, and you want to locate the end-points of the interval that you associate with the word on that scale, but you are unsure of these two end-points:* (*Q1*)[(*Q2*)] *On the scale of l to r, what are the endpoints of an* ***interval of numbers*** *that you associate with the left* [*right*] *end-point of the word?* A single subject provides $[a_L, b_L]$ and $[a_R, b_R]$ for each linguistic term. Explain how a collection of virtual subject data can be generated from $[a_L, b_L]$ and $[a_R, b_R]$.

References

Berenji, H. R. 1988. Treatment of uncertainty in artificial intelligence. In *Machine intelligence and autonomy aerospace aystems,* eds. H. Heer and H. Lum, pp. 233–247, AIAA, Washington, DC.

John, R., and S. Coupland. 2007. Type-2 fuzzy logic: A historical view. *IEEE Computational Intelligence Magazine* 2: 57–62.

Klir, G.J., and T.A. Folger. 1988. *Fuzzy sets, uncertainty, and information*. Englewood Cliffs, NJ: Prentice Hall.

Klir, G.J., and M.J. Wierman. 1998. *Uncertainty-based information*. Heidelberg, Germany: Physica-Verlag.

Korjani, M.M., and J.M. Mendel. 2013. *Fuzzy love selection by means of perceptual computing*, 766–770. Proc. IFSA World Congress and NAFIPS Annual Meeting: Edmonton, Canada, June.

Mendel, J.M. 1983. *Optimal seismic deconvolution: An estimation based approach*. New York: Academic Press.

Mendel, J.M. 1990. *Maximum-likelihood deconvolution: A journey into model-based signal processing*. New York: Springer-Verlag.

Mendel, J.M. 1995. *Lessons in estimation theory for signal processing*. Prentice-Hall PTR, Englewood Cliffs, NJ: Communications and Control.

Mendel, J. M. 1999. Computing with words, when words can mean different things to different people. In *Proceeding of third international ICSC symposium on fuzzy logic and applications*, June 1999, pp. 158–164. Rochester Institute of Technology, Rochester, NY.

Mendel, J. M. 2003. Fuzzy sets for words: A new beginning. In *Proceeding of IEEE international conference on fuzzy system*, May 26–28, pp. 37-42. St. Louis, MO.

Mendel, J.M. 2007a. Computing with words and its relationships with fuzzistics. *Information Sciences* 177: 998–1006.

Mendel, J.M. 2007b. Computing with words: Zadeh, Turing, Popper and Occam. *IEEE Computational Intelligence Magazine* 2: 10–17.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: Aiding people in making subjective judgments*. Hoboken, NJ: Wiley and IEEE Press.

Mendel, J. M. and D. Wu. 2014. Determining interval type-2 fuzzy set models for words using data collected from one subject: person FOUs. In *Proceeding of 2014 IEEE international conference on fuzzy systems* (FUZZ-IEEE), pp. 768–775, Beijing, China.

Türksen, I. 1986. Interval valued fuzzy sets based on normal forms. *Fuzzy Sets and Systems* 20: 191–210.

Türksen, I.B. 1992. Interval-valued fuzzy sets and compensatory. *Fuzzy Sets and Systems* 51: 295–307.

Wu, H., and J.M. Mendel. 2004. On choosing models for linguistic connector words for Mamdani fuzzy logic systems. *IEEE Transcation on Fuzzy Systems* 12: 29–44.

Zadeh, L.A. 1975. The concept of a linguistic variable and its application to approximate reasoning–1. *Information Sciences* 8: 199–249.

Zadeh, L.A. 1996. Fuzzy logic = computing with words. *IEEE Transcation on Fuzzy Systems* 4: 103–111.

Zadeh, L. A. 1999. From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions. In *IEEE transcation on circuits and systems–I: fundamental theory and applications*, vol. 4, pp. 105–119.

Zimmermann, H.J., and P. Zysno. 1980. Latent connectives in human decision making. *Fuzzy Sets and Systems* 4: 37–41.

Chapter 6
Type-2 Fuzzy Sets

6.1 The Concept of a Type-2 Fuzzy Set

Consider[1] the transition from ordinary sets to fuzzy sets. When the membership of an element in a set cannot be determined as either 0 or 1, type-1 fuzzy sets are used. Similarly, when the circumstances are so uncertain that it is difficult to determine the membership grade even as a crisp number in [0, 1] then fuzzy sets of type 2 can be used, a concept that was first introduced in Zadeh (1975).

When something is uncertain (e.g., a measurement), its exact value cannot be determined, so using type-1 sets makes more sense than using crisp sets. However, using exactly specified type-1 fuzzy set membership functions (MFs) seems counterintuitive—an early criticism of fuzzy sets. If the exact value of an uncertain quantity cannot be determined, then how can the exact membership of a fuzzy set be determined? Of course, this criticism applies to type-2 fuzzy sets as well, because even though their membership is fuzzy, it is still specified exactly, which again seems counterintuitive. Continuing to think along these lines, it follows that no finite-type fuzzy set can represent uncertainty "completely." So, ideally, a type-∞ fuzzy set has to be used to completely represent uncertainty. Of course, this cannot be done in practice, so some finite-type sets have to be used. This book deals just with type-1 and type-2 fuzzy sets. Higher type fuzzy sets could be examined, but the complexity of such fuzzy sets increases rapidly.

Uncertain MFs also occur when fuzzy sets are used as models for linguistic terms (words). When one wants to simultaneously model intra- and interlinguistic uncertainties (Sect. 5.2), this cannot be done using type-1 fuzzy sets, but it can be done by using type-2 fuzzy sets.

Example 6.1 The number of terms associated with a linguistic variable is free to be chosen. In Fig. 6.1 the linguistic variable temperature has been decomposed into five terms {*very low*, *moderately low*, *near zero*, *moderately high*, *very high*}.

[1]The first two paragraphs in this section are adapted from Karnik and Mendel (1998, p. 2).

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_6

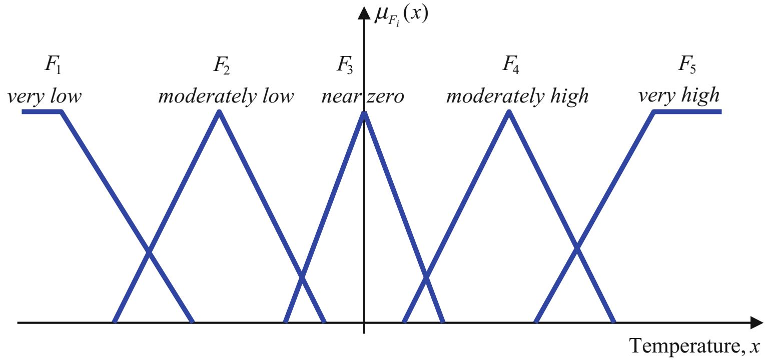


Fig. 6.1 Type-1 MFs

For illustrative purposes, triangular and piecewise linear MFs have been used. Many decisions were made to draw Fig. 6.1 MFs, including where to center the triangles, where to locate the shoulder points of the piecewise linear MFs, where to locate the basepoints of the triangles on the x-axis, and how much overlap there should be between neighboring MFs. All of these decisions translate into MF uncertainties, which, as shall be demonstrated in this chapter, can be handled by type-2 fuzzy sets and their MFs.

Example 6.2 This is a continuation of Example 2.2, so it would be a good time for the reader to reread that example. Now a model is sought that lets uncertainties about the overlap of the fuzzy sets be incorporated into it. A type-2 fuzzy set lets us do that and in different ways. In Fig. 6.2b, c, each of the type-1 overlap points [b, c, d, and e in (a)] is allowed to have an interval of uncertainty. For the purposes of this example, imagine a multitude of straight lines each with the same slope as the straight line type-1 MF, sweeping from the left end of the interval of uncertainty to the right end of that interval (other nonparallel lines, or even curves, could also be used). A couple of these lines are shown as dashed lines in Fig. 6.2b, c. In Fig. 6.2b the same weight (which leads to the uniform shading) is assigned to every point on each of these straight lines, in which case the resulting type-2 fuzzy set is one example of what is called an *interval type-2 fuzzy set*. In Fig. 6.2c the same weight is assigned to each line but only at a level u (which leads to the nonuniform shading), in which case the resulting type-2 fuzzy set is one example of what is called a *general type-2 fuzzy set*. Although the overlapping of x is the same for both kinds of type-2 fuzzy sets, the weightings (which occur in the third dimension) are different.

So, a type-2 fuzzy set lets one partition x using *overlapping partitions*, but now the partitions allow for uncertainty about the overlap, i.e., as *second-order uncertainty partitions* (Definition 1.3), something that cannot be done by a type-1 fuzzy set. Such overlapping partitions lead to even smoother transitions from one set to

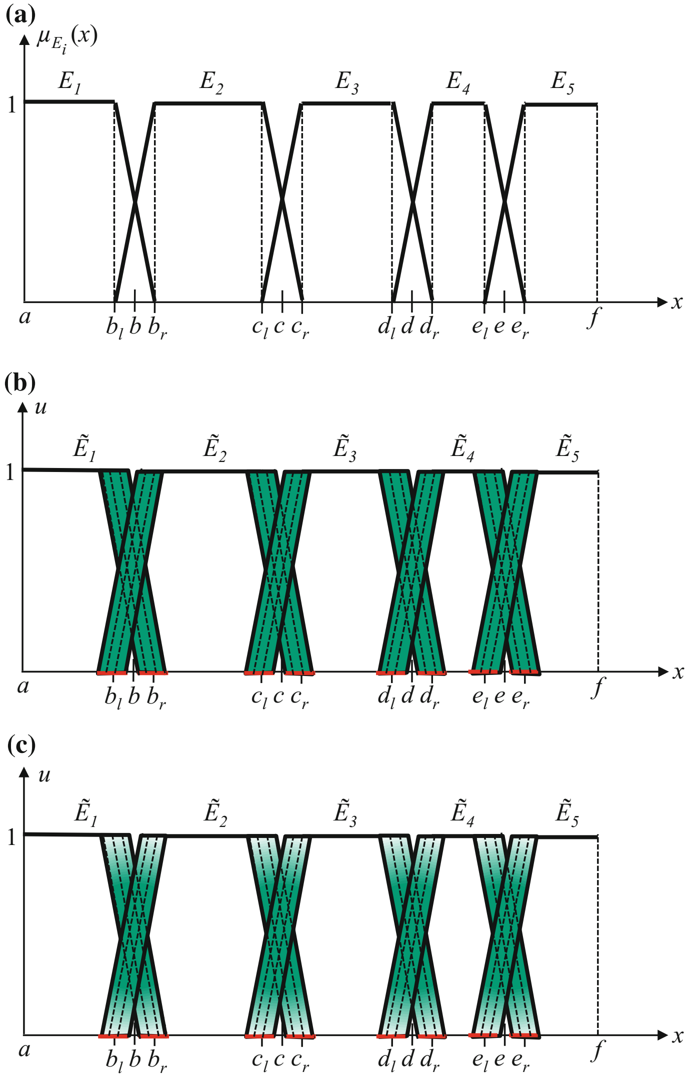


Fig. 6.2 Transitions from overlapping partitions for type-1 fuzzy sets to type-2 fuzzy sets: **a** interpreting type-1 fuzzy sets as overlapping partitions, **b** interval type-2 fuzzy set overlapping partitions (uniform shading), and **c** general type-2 fuzzy set overlapping partitions (nonuniform shading). In (**b**) and (**c**) the type-1 overlapping partition points (e.g., b_l and b_r) have expanded to intervals, $b_l \to [b_{ll}, b_{lr}]$ and $b_r \to [b_{rl}, b_{rr}]$, respectively. The endpoints of these intervals are not marked in (**b**) or (**c**)

another than the type-1 fuzzy set transitions, and so they have the potential to serve us well where it is important to allow for uncertainty about the overlap.

Additional benefits for using general type-2 fuzzy sets are described in Sect. 11.1.

Finally, the just-used phrase "second-order uncertainty partitions" may sound to some as diminishing the importance of a type-2 fuzzy set. It will be seen that a type-2 fuzzy system is analogous to a probabilistic system through first- and second-order moments, whereas a type-1 fuzzy system is analogous to a probabilistic system only through the first moment. And, of course, type-2 fuzzy sets are needed to implement a type-2 fuzzy system. Exactly how this is done is the subject of Chaps. 9 and 11.

Zadeh (1975, Definition 3.1) defined a type-2 fuzzy set as one whose MF ranges over type-1 fuzzy sets, but he left open how to mathematically model such fuzzy sets. The rest of this chapter explains some ways for doing this.

6.2 Definitions of a General Type-2 Fuzzy Set and Associated Concepts

Imagine blurring the type-1 MF depicted in Fig. 6.3a by shifting the points on the triangle either to the left or to the right and not necessarily by the same amounts, as in Fig. 6.3b. Then, at a specific value of x, say x', there no longer is a single value for the MF; instead, the MF takes on values wherever the vertical line intersects the blur. Those values need not all be weighted the same; hence, an amplitude distribution can be assigned to all of those points. Doing this for $x \in X$, a three-dimensional MF—a type-2 MF—is created that characterizes a type-2 fuzzy set. This construction can be easily remembered, as *type-1-blur-weight*.

John and Coupland (2012) make the following important cautionary statement about "blurring":

> A careful rereading of [the above paragraph] will demonstrate that the notion of blurring is used to illustrate and communicate the form for a general type-2 fuzzy MF and less so an advocated methodology for constructing a type-2 MF. In some applications blurring may work well (Coupland and John 2007), however, there is no guarantee of success and in certain application areas the blurring approach will fail.

Since the publication of the first edition of this book (Mendel 2001) some of the notations and definitions about type-2 fuzzy sets have changed for the better. A history of those changes, as well as recommended notation and definition changes appears in Mendel et al. (2016). This book abides by all of those recommended changes.

Definition 6.1 A *type-2 fuzzy set* (also called a *general type-2 fuzzy set*), denoted $\tilde{A}$, is the graph of a bivariate function (Aisbett et al. 2010)—called the MF of $\tilde{A}$—on the Cartesian product $X \times [0,1]$ into $[0,1]$, where X is the universe for the *primary variable* of $\tilde{A}$, x. The MF of $\tilde{A}$ is denoted $\mu_{\tilde{A}}(x,u)$, (or $\mu_{\tilde{A}}$ for short) and is called a *type-2 MF*, i.e.,

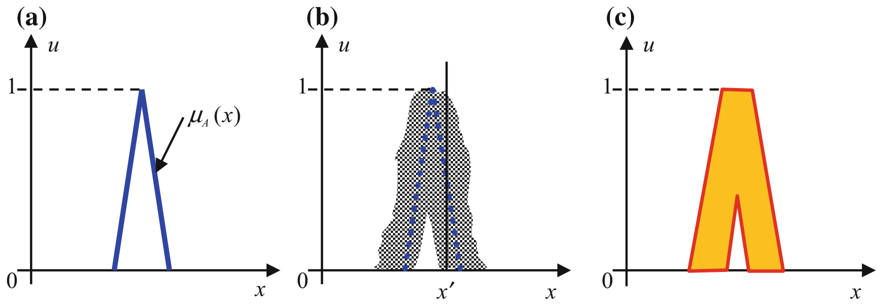


Fig. 6.3 **a** Type-1 MF, **b** blurred type-1 MF (artistic rendition), and **c** footprint of uncertainty

$$\tilde{A} = \left\{ \left((x,u), \mu_{\tilde{A}}(x,u) \right) | x \in X,\ u \in U \equiv [0,1] \right\} \tag{6.1}$$

in which $0 \le \mu_{\tilde{A}}(x,u) \le 1$. U is the universe for the *secondary variable* u, and in this book U is always assumed to be[2] $[0,1]$. $\tilde{A}$ can also be expressed in fuzzy set notation as

$$\tilde{A} = \int_{x \in X} \int_{u \in [0,1]} \mu_{\tilde{A}}(x,u)/(x,u), \tag{6.2}$$

where $\iint$ denotes union over all admissible x and u.[3]

In (6.1), the restriction $u \in U \equiv [0,1]$ is consistent with the type-1 constraint that $0 \le \mu_A(x) \le 1$, that is, when MF uncertainties disappear, a type-2 MF must reduce to a type-1 MF, in which case the variable u equals $\mu_A(x)$ and $0 \le \mu_A(x) \le 1$. The restriction $0 \le \mu_{\tilde{A}}(x,u) \le 1$ is consistent with the fact that the amplitude of a MF should lie between or be equal to 0 and 1.

Unless otherwise stated, in this book:

1. The elements of X and U are assumed to be continuous real numbers. Where needed, their discrete versions are denoted X_d and U_d, respectively.
2. If one begins with continuous X and U and discretizes (samples) them for computing purposes, then X and U will continue to be viewed as continuous, because the original type-2 fuzzy sets were defined on continuous X and U.

Note that $\mu_{\tilde{A}}(x,u)$, whose domain is $X \times U$, is three-dimensional [just as $\mu_A(x)$ for a type-1 fuzzy set is two-dimensional]. Consequently, regardless of what kind of a type-2 fuzzy set one may be describing, for it to be called a type-2 fuzzy set its MF must be three-dimensional.

---

[2]If U differs from $[0,1]$, then use $u \in U$ in (6.1), and do the same elsewhere, e.g. in (6.4a) and (6.4c).

[3]For discrete universes of discourse, in (6.2) $\int$ is replaced by $\sum$, X is replaced by X_d and $[0,1]$ is replaced by $\{0, u_1, u_2, \ldots, u_{n-1}, 1\}$.

Example 6.3 Figure 6.4 depicts a 3-D MF for a general type-2 fuzzy set. Observe that, for this example, primary variable $x \in [0, 1]$, but in general X is application dependent, whereas U is always $[0, 1]$.

Definition 6.2 For every $x \in X$ and $u \in [0, 1]$, the value of $\mu_{\tilde{A}}(x, u), f_x(u)$, is called the *secondary grade* of x; hence, if $x' \in X$ and $u' \in [0, 1]$, then $f_{x'}(u') \equiv \mu_{\tilde{A}}(x', u')$ where $0 \leq f_{x'}(u') \leq 1$.

Definition 6.3 A *secondary membership function*, $\mu_{\tilde{A}(x)}(u)$, is (Aisbett et al. 2010) a restriction of function $\mu_{\tilde{A}} : X \times [0, 1] \to [0, 1]$ to $x \in X$, i.e., $\mu_{\tilde{A}(x)} : [0, 1] \to [0, 1]$, or in fuzzy set notation:

$$\mu_{\tilde{A}(x)}(u) = \int_{u \in [0,1]} \mu_{\tilde{A}}(x, u)/u = \int_{u \in [0,1]} f_x(u)/u \tag{6.3}$$

Note, importantly, that $\tilde{A}(x)$ is a type-1 fuzzy set,[4] which is also referred to as a *secondary set*, and as such it can be represented by its $\alpha-$ cut decomposition (see Sect. 2.12). $\{\mu_{\tilde{A}(x)}(u) | u \in U\}$ is also called a *vertical-slice* of $\mu_{\tilde{A}}(x, u)$.

A viable alternative to symbol $\tilde{A}(x)$ is $\tilde{A}_x$. The latter is occasionally used in later chapters where parenthetical arguments are needed for other things.

If all of the secondary MFs have the same geometric shape (e.g., triangle, trapezoidal, Gaussian, etc.), then the name of that geometric shape is used as the name for the entire type-2 fuzzy set.

Because a secondary MF is a type-1 fuzzy set it can be convex or non-convex (see Definition 2.4). Almost all published results for type-2 fuzzy sets are for convex secondary MFs; however, beginning in the year 2010 (Tahayori et al. 2010), interest in non-convex secondary MFs awakened. As of the writing of this book, there are too many unanswered theoretical and computational issues associated with using non-convex secondary MFs (more will be said about this in Chap. 7); hence, unless otherwise stated, *in this book secondary MFs will always be convex functions*.

Example 6.4 Figure 6.5 depicts triangle secondary MFs at five nonzero values of the primary variable $(x = x_1, \ldots, x_5)$ and spike secondary MFs at $x = 0$ and $x = x_6$. Triangle secondary MFs are shown only because they are easy to draw. All of the secondary MFs have one maximum membership grade equal to 1, and so they are *normal* type-1 fuzzy sets. Why there are spike secondary MFs at $x = 0$ and $x = x_6$ will be explained in Example 6.8. Finally, because the secondary MFs are triangles (spikes can be interpreted as triangles) this type-2 fuzzy set is called a *triangle type-2 fuzzy set*, whereas the fuzzy set in Fig. 6.4 is called a *Gaussian type-2 fuzzy set*, because all of its secondary MFs are Gaussians.

---

[4]Notation $\mu_{\tilde{A}(x)}(u)$ [which is different from the notation used in Mendel (2001)] is consistent with the usual labeling of a MF for a type-1 fuzzy set, where $\tilde{A}(x)$ is the name of that set.

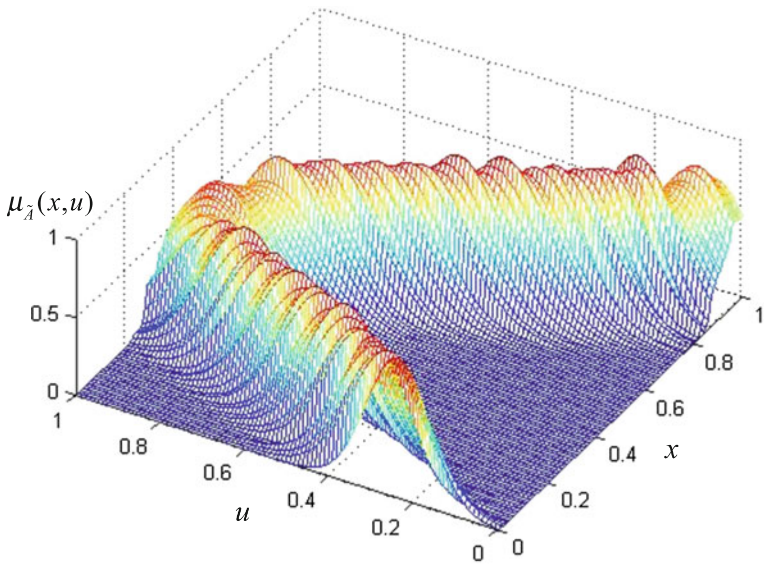


Fig. 6.4 A 3-D MF for a general type-2 fuzzy set (courtesy of Prof. Frank Rhee)

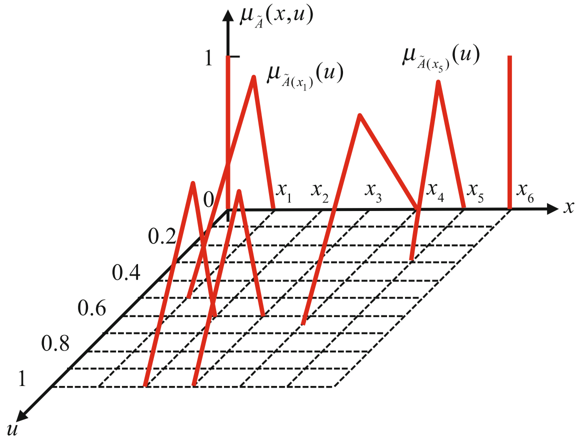


Fig. 6.5 Secondary MFs (in *red*) at seven values of the primary variable

Example 6.5 This is a continuation of Example 6.4. Shown in Fig. 6.6 are the nonzero secondary grades at the Fig. 6.5 grid points of $X \times U$. A few of the nonzero secondary grades are labeled, e.g. $f_{x_1}(0.1), f_{x_4}(0.3)$ and $f_{x_6}(0)$.

Secondary MFs can be chosen in different ways. In Examples 6.4 and 6.5 they were chosen a priori as triangles. Triangle, trapezoid, truncated Gaussian, etc., secondary MFs are very commonly used during designs of type-2 fuzzy systems when all MF parameters are optimized.

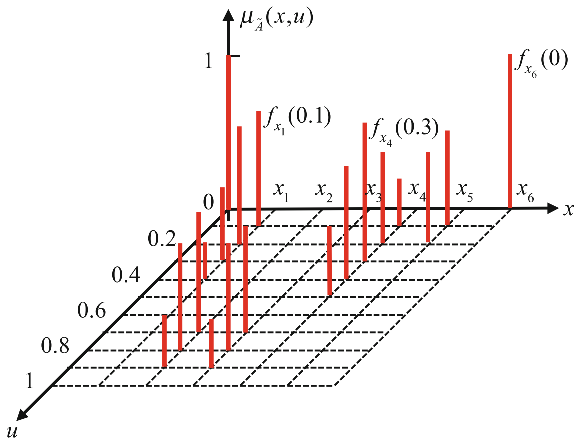


Fig. 6.6 Nonzero secondary grades at the grid points of $X \times U$ for the seven secondary MFs that are depicted in Fig. 6.5

Secondary MFs can also be obtained directly from data, e.g., Moharrer et al. (2013) collect data about words that are then processed using statistics-based techniques that include confidence intervals that vary between 0 and 1, which then play the role of secondary grades; and, Bilgin et al. (2012a, b, c, 2013a, b) begin with a linguistic term that is modeled as either a left or right shoulder, apply n concentration hedges to it [the kind that make the membership of the hedged word contained within the membership of its less-hedged word (Sect. 2.9)], and then layer the hedged FOUs one on top of another at secondary grades equal to integer multiples of $1/n$. This *linear adjective* model is so far limited to shoulder models. Rakshit et al. (2013, 2016) solve an optimization problem to determine secondary MFs that characterize the reliability in the assignment of the primary membership using subject data.

Definition 6.4 J_x, the *primary membership* of x, is[5]:

$$J_x = \{(x,u)|u \in [0,1],\ \mu_{\tilde{A}}(x,u) > 0\} \tag{6.4a}$$

It can also be expressed as a subset of $\{x\} \times I_x$ (Mendel et al. 2016), i.e.,

$$J_x = \{x\} \times I_x, \tag{6.4b}$$

where

$$I_x = \{u \in [0,1]|\mu_{\tilde{A}}(x,u) > 0\} \tag{6.4c}$$

[5]In Mendel (2001) and much of the type-2 literature it is stated that $J_x \subseteq [0,1]$ and J_x is left undefined, both of which have been very problematic. In this book, the statement $J_x \subseteq [0,1]$ has been abandoned and J_x is defined. For additional discussions about this, see Sect. 6.6 and Mendel et al. (2016).

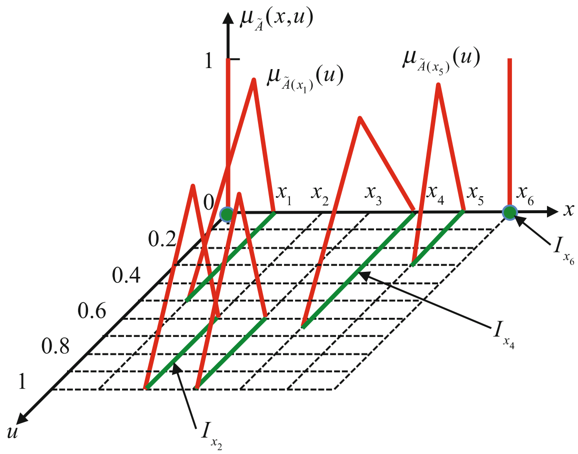


Fig. 6.7 Supports of the secondary MFs are shown in *green* for the seven secondary MFs that are depicted in Fig. 6.5

I_x is the *support of the secondary MF*, and can be *connected*[6] or *disconnected*.

Example 6.6 This is a further continuation of Example 6.4. Shown in Fig. 6.7 are the supports of the secondary MFs at the seven values of the primary variable that are depicted in Fig. 6.5. A few of the supports are labeled (e.g., I_{x_2}, I_{x_4}, and I_{x_6}). Observe that at $x = 0$ and $x = x_6$, I_0 and I_{x_6} are single points, whereas I_{x_1}, I_{x_2}, I_{x_3}, I_{x_4} and I_{x_5} are intervals. In this example each I_x is *connected*. An example where this is not so is given in Example 6.7.

Definition 6.5 (Mendel et al. 2016) The *support* of $\tilde{A}$ (Aisbett et al. 2010) comprises all $(x, u) \in X \times [0, 1]$ such that $\mu_{\tilde{A}}(x, u) > 0$, and is also called the *domain of uncertainty* of $\tilde{A}$, $\mathrm{DOU}(\tilde{A})$, i.e.,

$$\mathrm{DOU}(\tilde{A}) = \{(x, u) \in X \times [0, 1] | \mu_{\tilde{A}}(x, u) > 0\} = \bigcup_{x \in X} J_x. \tag{6.5}$$

It is worth noting (Mendel et al. 2016) that the word "uncertainty" in "DOU" refers to the uncertainty that is expressed by the secondary MF; therefore, one might argue that $\mathrm{DOU}(\tilde{A})$ should be even more restricted than above, i.e., it should be defined by removing isolated points in the support that have unity secondary membership grades. This is not done because "DOU" refers to a collective entity, and so even if there is only one element in it that has an uncertain membership grade it can be called a "DOU."

Example 6.7 Shown in Fig. 6.8 are the secondary MFs for a general type-2 fuzzy set at five values of the primary variable. Observe that each of the supports of the secondary MFs, shown in green, is *disconnected*. $\mathrm{DOU}(\tilde{A})$ is comprised of all the green supports of the secondary MFs, which also corresponds collectively to the union of all of the primary memberships, as in (6.5).

---

[6]A set $A \subseteq \mathbb{R}$ is *connected* if and only if A is an interval (closed, open, or neither).

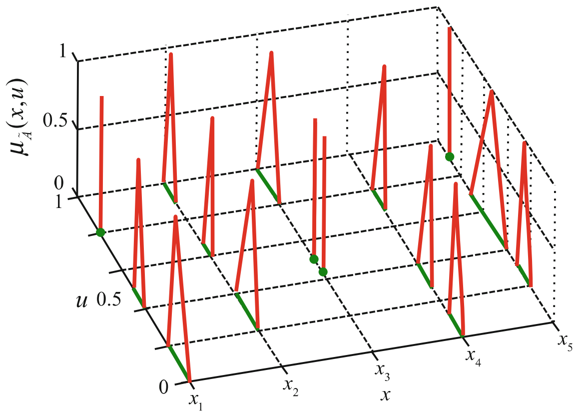


Fig. 6.8 The domain of uncertainty is shown in *green*

Definition 6.6 When the support of the secondary MF, I_x, is *closed* (so that it is connected; see footnote 6) for $x \in X$, i.e.,

$$I_x = \{u \in [0,1] | \mu_{\tilde{A}}(x,u) > 0\} = [\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x)], \tag{6.6}$$

where (Aisbett et al. 2010)[7]

$$\overline{\mu}_{\tilde{A}}(x) = \sup\{u | u \in [0,1], \mu_{\tilde{A}}(x,u) > 0\} \tag{6.7}$$

$$\underline{\mu}_{\tilde{A}}(x) = \inf\{u | u \in [0,1], \mu_{\tilde{A}}(x,u) > 0\}, \tag{6.8}$$

then the domain of uncertainty of $\tilde{A}$ is called the *footprint of uncertainty*[8] of $\tilde{A}$, $\text{FOU}(\tilde{A})$, i.e.,

$$\text{DOU}(\tilde{A}) = \text{FOU}(\tilde{A}) = \{(x,u) | x \in X \text{ and } u \in [\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x)]\}. \tag{6.9}$$

Note that $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$ are called the *lower* and *upper MFs*of $\text{FOU}(\tilde{A})$ (Mendel and Liang 1999), and are the lower and upper (type-1 fuzzy set) bounding functions

[7]Aisbett et al. (2010) were the first to express $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$ as in (6.7) and (6.8) and to point out that $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$ are the infimum and supremum of the support of $\mu_{\tilde{A}}$ expressed as functions on X. See Mendel et al. (2016) for additional discussions about this.

[8]The term *footprint of uncertainty* is not mentioned in Karnik and Mendel (2001b), and is defined only using words in Karnik and Mendel (2001a, Example 1), Karnik et al. (1999, Example 3.1), Liang and Mendel (2000, Definition 1), and Mendel (2007, Box 1). It was coined by Mendel as a simple way to verbalize and describe the two-dimensional domain of support for a type-2 fuzzy set's MF, and appears for the first time in Karnik et al. (1999) and Karnik and Mendel (2001a). Because the phrase *footprint of uncertainty* is not general enough to accommodate all kinds of type-2 fuzzy sets, the phrase *domain of uncertainty* was introduced in Mendel and John (2002). The concepts of *lower* and *upper MFs* were first described in Liang and Mendel (2000).

of the FOU, respectively. Commonly used abbreviations for the lower and upper MFs of $\tilde{A}$ are LMF$(\tilde{A})$ and UMF$(\tilde{A})$.

The definition of FOU$(\tilde{A})$ in (6.9) guarantees that it is connected if X is connected and if LMF$(\tilde{A})$ and UMF$(\tilde{A})$ are continuous functions $X \to [0,1] \subseteq \mathbb{R}$.

Definition 6.7 The *support* of LMF$(\tilde{A})$ [UMF$(\tilde{A})$] is the crisp set of all points $x \in X$ such that $\underline{\mu}_{\tilde{A}}(x) > 0$ [$\overline{\mu}_{\tilde{A}}(x) > 0$].

In general, the supports of LMF$(\tilde{A})$ and UMF$(\tilde{A})$ are different, and the support of LMF$(\tilde{A})$ is contained within the support of UMF$(\tilde{A})$.

Example 6.8 This is a further continuation of Example 6.4. Shown in Fig. 6.9 is the shaded-in FOU associated with the secondary MFs that are depicted in Fig. 6.5. In fact, Fig. 6.5 was actually obtained after Fig. 6.9 was constructed, by deleting the FOU in Fig. 6.9.

It should now be clear why there are spike secondary MFs at $x = 0$ and $x = x_6$; the secondary MFs at $x = 0$ and $x = x_6$ are single points.

Observe in Fig. 6.9 that both LMF$(\tilde{A})$ and UMF$(\tilde{A})$ are trapezoids, and that UMF$(\tilde{A})$ is a normal type-1 fuzzy set, whereas LMF$(\tilde{A})$ is a subnormal type-1 fuzzy set. In general, the LMF or the UMF of an FOU does not have to be a normal type-1 fuzzy set. In fact, in a type-2 fuzzy system in which fired rules are first combined by using the union operation the resulting FOU will not have normal lower and upper MFs. This will be demonstrated in Chap. 9.

Observe, also, that the support of LMF$(\tilde{A})$ is $[x_1, x_4]$, the support of UMF$(\tilde{A})$ is $[0, x_6]$, and $[x_1, x_4] \subset [0, x_6]$.

Definition 6.8 For continuous universes of discourse X and $U = [0,1]$, an[9] *embedded type-2 fuzzy set*, denoted $\tilde{A}_e$, is (Aisbett et al. 2010) $\mu_{\tilde{A}}$ restricted to $\{(x, u(x)) | x \in X\}$ for some $u : X \to [0,1]$, i.e., $\mu_{\tilde{A}_e} : X \to [0,1]$ when $\mu_{\tilde{A}_e}(x) = \mu_{\tilde{A}}(x, u(x))$, or in fuzzy set notation (Karnik and Mendel 1998)[10]:

$$\tilde{A}_e = \int_{x \in X} [f_x(u)/u]/x \quad u \in [0,1] \tag{6.10}$$

Set $\tilde{A}_e$ is embedded in set $\tilde{A}$, and there are an uncountable number of embedded type-2 fuzzy sets.

[9] *Embedded type-2* and *type-1 fuzzy sets* were first described in Karnik and Mendel (1998).

[10] In Mendel (2001) and many later references, (6.10) is stated for $u \in J_x$ instead of $u \in [0,1]$. While this is okay for connected primary memberships, it is not okay for disconnected primary memberships, because J_x is undefined when $\mu_{\tilde{A}}(x, u) = 0$ (see Definition 6.4). Using $u \in [0,1]$ in (6.10) remedies this. The same is true for (6.13).

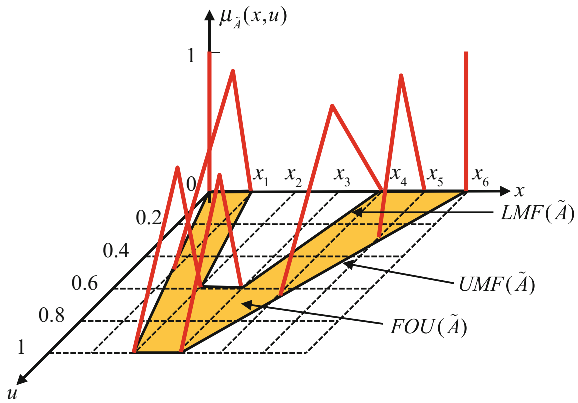


Fig. 6.9 The FOU for the secondary MFs that are depicted in Fig. 6.5

Definition 6.9 Given the closed I_x in (6.6), when $x = x_i$ and u is *discretized*, $[\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]$ becomes a set with $M_i + 1$ elements, $\{u_j^i\}_{j=1}^{M_i+1}$, defined herein as:

$$u_j^i \equiv \underline{\mu}_{\tilde{A}}(x_i) + (j-1) \times \left[\frac{\overline{\mu}_{\tilde{A}}(x_i) - \underline{\mu}_{\tilde{A}}(x_i)}{M_i - 1}\right], \quad j = 1, \ldots, M_i \tag{6.11}$$

This I_x is denoted I_x^D to signify it is for discretized u.

Definition 6.10 When X and U are discretized and I_x are closed, an *embedded type-2 fuzzy set* $\tilde{A}_e^j$ has N elements, one each from $I_{x_1}^D, I_{x_2}^D, \ldots, I_{x_N}^D$, namely $u_j^1, u_j^2, \ldots, u_j^N$, where u_j^i is defined in (6.11); and, each element has an associated secondary grade, namely $f_{x_1}(u_j^1), f_{x_2}(u_j^2), \ldots, f_{x_N}(u_j^N)$, i.e., in fuzzy set notation:

$$\tilde{A}_e^j = \sum_{i=1}^{N} \left[f_{x_i}(u_j^i)/u_j^i\right]/x_i \tag{6.12}$$

Set $\tilde{A}_e^j$ is embedded in $\tilde{A}$, and, there are a total of $\prod_{i=1}^{N} M_i\ \tilde{A}_e^j$.

Of course, u can be discretized in ways that are different from (6.11),[11] in which case the formula for u_j^i would change but the conceptual idea of an embedded type-2 set would not change.

Definition 6.11 For continuous universes of discourse X and U, an *embedded type-1 fuzzy set*, denoted A_e, is (Aisbett et al. 2010) a function whose range is a subset of $[0,1]$ determined by $\mu_{\tilde{A}}$, i.e., $u : X \to [0,1]$ satisfying $\mu_{\tilde{A}}(x, u(x)) > 0$ for $x \in X$, or in fuzzy set notation (Karnik and Mendel 1998):

[11] Equation (6.11) leads to uniform sampling at each x_i. A *grid method* of discretization is given in Greenfield and John (2007), in which x and u are "evenly divided into a rectangular grid, as determined by the degree of discretization of the x and u axes".

$$A_e = \int_{x \in X} u/x \quad u \in [0, 1] \tag{6.13}$$

Set A_e acts as the domain for $\tilde{A}_e$ in (6.10), and there are an uncountable number of embedded type-1 sets.

Definition 6.12 When X and U are discretized and I_x are closed, an *embedded type-1 fuzzy set* A_e^j has N elements, one each from $I_{x_1}^D, I_{x_2}^D, \ldots, I_{x_N}^D$, namely $u_j^1, u_j^2, \ldots, u_j^N$, where u_j^i is defined in (6.11), i.e., in fuzzy set notation:

$$A_e^j = \sum_{i=1}^{N} u_j^i / x_i \tag{6.14}$$

Set A_e^j acts as the domain for $\tilde{A}_e^j$ in (6.12), and there are a total of $\prod_{i=1}^{N} M_i \, A_e^j$.

Example 6.9 This is a further continuation of Example 6.5, in which the grid on $X \times U$ defines $X_d \times U_d$. Observe from Fig. 6.6 that there are $\prod_{i=0}^{6} M_i = 1 \times 4 \times 3 \times 3 \times 5 \times 2 \times 1 = 360$ embedded type-2 and type-1 fuzzy sets. Figure 6.10 depicts two of the embedded type-2 and type-1 fuzzy sets. The former are shown in red and the latter are shown in green. For Fig. 6.10a:

$$\begin{cases} A_e = 0/0 + 0.4/x_1 + 0.8/x_2 + 0.9/x_3 + 0.2/x_4 + 0.1/x_5 + 0/x_6 \\ \tilde{A}_e = 1/0/0 + 0.25/0.4/x_1 + 0.6/0.8/x_2 + 0.3/0.9/x_3 + 0.6/0.2/x_4 + 0.6/0.1/x_5 + 1/0/x_6 \end{cases} \tag{6.15a}$$

For Fig. 6.10b:

$$\begin{cases} A_e = 0/0 + 0.1/x_1 + 0.9/x_2 + 0.7/x_3 + 0.5/x_4 + 0.2/x_5 + 0/x_6 \\ \tilde{A}_e = 1/0/0 + 0.75/0.1/x_1 + 0.3/0.9/x_2 + 0.75/0.7/x_3 + 0.45/0.2/x_4 + 0.55/0.2/x_5 + 1/0/x_6 \end{cases} \tag{6.15b}$$

Definition 6.13 Assume that each of the secondary MFs of a type-2 fuzzy set has only one secondary grade that equals 1. A *principal MF*is the union of all such points at which this occurs, i.e.,

$$\mu_{\text{principal}}(x) = \int_{x \in X} u/x \quad \text{where } f_x(u) = 1 \tag{6.16}$$

and[12] is associated with a type-1 fuzzy set.

[12]Note that $f_x(u) = 1$ can be solved for u (at $x \in X$), so that u can be expressed as $u = f_x^{-1}(1)$.

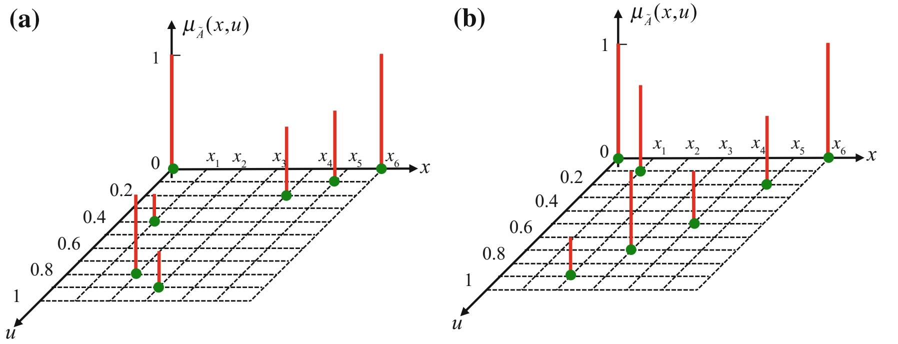
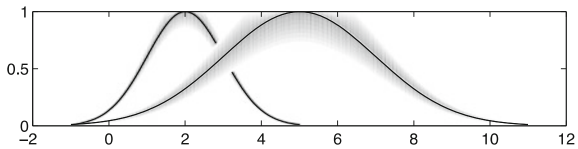


Fig. 6.10 Two examples of (*red*) embedded type-2 and (*green*) type-1 fuzzy sets, each associated with the sampled type-2 fuzzy set that is depicted in Fig. 6.6

Fig. 6.11 Pictorial representation of a Gaussian type-2 fuzzy set



Example 6.10 The principal MFs for the two Gaussian type-2 fuzzy sets whose MFs are depicted in Fig. 6.11 are the solid Gaussian curves.

Definition 6.14 The MF of a type-2 fuzzy set is said to be *parsimonious* when it is described by a small number of parameters.

Parsimonious models have a very long history in the field of mathematical modeling, e.g., in system identification [e.g., Ljung (1999)] one always tries to use a model with the fewest number of parameters to fit data. Because a type-2 fuzzy set is a mathematical model, parsimony should be adhered to for such models. Justification for preferring a parsimonious model seems to go all the way back to Occam.[13]

[13]William of Ockham (also Occam) (c. 1288–1348) was an English Franciscan friar and scholastic philosopher, from Ockham, a small village in Surrey, near East Horsley. One important contribution that he made to modern science and modern intellectual culture was through the principle of parsimony in explanation and theory building what came to be known as Ockham's razor. This maxim, as interpreted by Bertrand Russell, states that if one can explain a phenomenon without assuming this or that hypothetical entity, there is no ground for assuming it, i.e., one should always opt for an explanation in terms of the fewest possible number of causes, factors, or variables. The most useful statement of the Ockham’s razor principle is “when you have two competing theories which make exactly the same predictions, the one that is simpler is the better.” This principle is sometimes misstated as “keep it as simple as possible.” One can have two (or more) competing theories that lead to different predictions. Occam’s Razor does not apply in that case, because the results that are obtained from the competing theories are different.

Although parsimony is advocated in this book, this does not mean there can be no flexibility about it, e.g., if a type-2 fuzzy set that is described by n_1 parameters does not lead to improved performance for an application, a type-2 fuzzy set model that is described by $n_1 + 1$ parameters should then be tried, etc.

Definition 6.15 A *totally symmetrical type-2 fuzzy set* is one whose lower and upper MFs, as well as its secondary MFs, are symmetrical.

Definition 6.16 (Pedrycz 2015) An *information granule* is a collection of elements drawn together by their closeness (resemblance, proximity, functionality, etc.), articulated in terms of some useful spatial, temporal, or functional relationship.

Clearly, a type-2 fuzzy set is an information granule (as is a type-1 fuzzy set). So, for example, when a word is modeled by a type-2 fuzzy set, this can be called a *granular model of a word*.

In the rest of this chapter and book, type-2 fuzzy set, interval type-2 fuzzy set and type-1 fuzzy set are often abbreviated to T2 FS, IT2 FS and T1 FS, respectively.

6.3 Definitions of an IT2 FS and Associated Concepts

The most widely studied T2 FS has all of its secondary grades equal to 1 and is called an IT2 FS.

Definition 6.17 When $u \in [0, 1]$ and $\mu_{\tilde{A}}(x, u) = 1$ for $x \in X$, then $\tilde{A}$ is called an *IT2 FS*. An IT2 FS is completely described by its DOU, so that:

$$\tilde{A} = 1/\text{DOU}(\tilde{A}) \tag{6.17}$$

(6.17) is an expressive equation that means $\mu_{\tilde{A}}(x, u) = 1$ for $(x, u) \in \text{DOU}(\tilde{A})$, where $\text{DOU}(\tilde{A})$ is given in (6.5) in which $\mu_{\tilde{A}}(x, u) > 0$ is replaced by $\mu_{\tilde{A}}(x, u) = 1$.

Equation (6.17) does not mean that one should ignore the secondary grades, because doing so would reduce the IT2 FS from a 3-D entity to a 2-D entity and so it would no longer be a T2 FS. All of the earlier definitions in this chapter can be used for an IT2 FS simply by setting the secondary grades equal to 1, and so they are not repeated here.

An IT2 FS is said to be *maximally uncertain* because all of its secondary membership grades are the same value. A general T2 FS is said to be *less uncertain than an IT2 FS* because its secondary grades are not all the same. How to provide a quantitative measure of uncertainty for a T2 FS is discussed in Chap. 8.

Definition 6.18 (Mendel et al. 2016) An IT2 FS is called a *closed IT2 FS* (CIT2 FS) when I_x is closed for $x \in X$ (see Definition 6.6). In this case, $\mathrm{DOU}(\tilde{A}) = \mathrm{FOU}(\tilde{A})$; hence, for a CIT2 FS, (6.17) can be expressed as:

$$\tilde{A} = 1/\mathrm{FOU}(\tilde{A}) \tag{6.18}$$

where $\mathrm{FOU}(\tilde{A})$ is defined in (6.9). Alternate ways to express $\mathrm{FOU}(\tilde{A})$ for a CIT2 FS are [see, also, (6.5)]:

$$\mathrm{FOU}(\tilde{A}) = \bigcup_{x \in X} \{x\} \times I_x = \bigcup_{x \in X} J_x \tag{6.19}$$

The vertical-slice of a CIT2 FS is a type-1 interval fuzzy number on U (Definition 2.6).

The CIT2 FS has received the most attention to-date, and is the one that is emphasized in this book (more about this in Sect. 6.6). Such an IT2 FS is related to an *interval-valued fuzzy set* (IVFS).

Definition 6.19 (Bustince et al. 2015) An *IVFS A* on the universe $X \neq \varnothing$ is a mapping[14]

$$A : U \to L([0,1]) = \left\{ \mathbf{x} = [\underline{x}, \bar{x}] | (\underline{x}, \bar{x}) \in [0,1]^2 \text{ and } \underline{x} \leq \bar{x} \right\} \tag{6.20}$$

such that the membership degree of $x \in X$ is given by $A(x) = [\underline{A}(x), \overline{A}(x)] \in L([0,1])$, where $\underline{A} : X \to [0,1]$ and $\overline{A} : X \to [0,1]$ are mappings defining the lower and upper bound of the membership interval $A(x)$, respectively.

Observe that no secondary grade is assigned to an IVFS and so it is not the same as an IT2 FS. *It is* $\mathrm{FOU}(\tilde{A})$ *that is an IVFS.*[15] More will be said about IT2 FSs and IVFSs in Sect. 6.6.

---

[14]The statement of this definition uses the notation in Bustince et al. (2015). To connect that notation with the notation that is used in this book, set $A(x) = I_x$, $\underline{A}(x) = \underline{\mu}_{\tilde{A}}(x)$ and $\bar{A}(x) = \bar{\mu}_{\tilde{A}}(x)$. There are also other notations that are used for an IVFS, e.g. (1) Goralczany (1987) calls them an i–v fuzzy set, where $\bar{\mu}_A(x) = [\mu_A^L(x), \mu_A^U(x)]$ ($\bar{\mu}_A$ does not denote an UMF); (2) Bustince (2000) denotes the degree of membership of an element x of an IVFS A as $M_A(x) = [M_{AL}(x), M_{AU}(x)]$; and, (3) Klir and Yuan (1995) denote the degree of membership of an element x of an IVFS A as $A(x) = [L_A(x), U_A(x)]$.

[15]Mendel et al. (2016, p. 341) states: "… every closed IT2 FS is an IVFS." This is not quite correct. An IT2 FS has always had a secondary grade associated with it, whereas an IVFS has not. Put a different way, the MF of a closed IT2 FS is three-dimensional whereas the MF of an IVFS is two-dimensional. An analogy can be made between these two kinds of fuzzy sets and a rectangular skyscraper. The IVFS is analogous to the foundation of this building, whereas the closed IT2 FS is analogous to the entire rectangular building. Clearly, the foundation cannot claim to be the entire building. The distinction between an IVFS and a closed IT2 FS is not very important when one is only interested in studying (closed) interval type-2 fuzzy systems because the unity secondary grades of the closed IT2 FSs convey no useful information; it is the FOU of the closed IT2 FS that plays the important role, and, as just stated, the FOU is the same as an IVFS; it is in this sense that the above quoted phrase should be

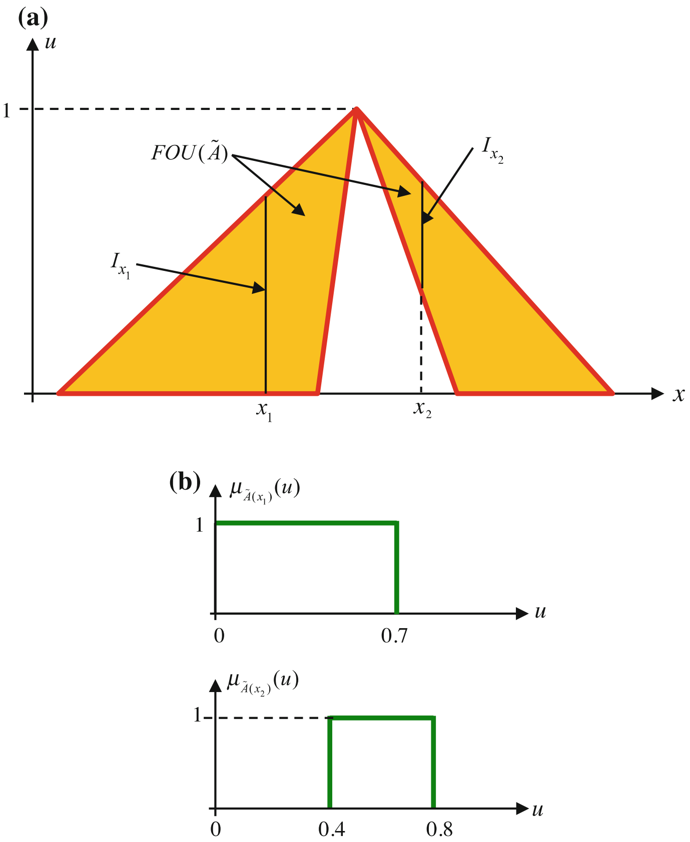


Fig. 6.12 The shaded region in **a** is the FOU of a CIT2 FS. The supports of the secondary MFs, I_{x_1} and I_{x_2}, are shown at x_1 and x_2 in (**a**), and their associated secondary MFs, $\mu_{\tilde{A}(x_1)}(u)$ and $\mu_{\tilde{A}(x_2)}(u)$, are shown in (**b**)

FOUs can be chosen in different ways. For CWW applications, data can be collected from a group of subjects about a word (as already explained in Sects. 4.4 and 5.2) after which that data is mapped into an FOU (more about this is explained in Sect. 10.4). For non-CWW applications, a type-1 MF may be blurred to obtain an FOU, or (more commonly) parsimonious parametric UMFs and LMFs are chosen a priori to define an FOU (e.g., triangle, trapezoid, Gaussian, etc.); their parameters are either prespecified or optimized during a design procedure (more about this is explained in Sect. 10.2).

(Footnote 15 continued)

interpreted. However, the distinction between an IVFS and a closed IT2 FS is very important when one studies general T2 FSs, for which the secondary grades are no longer all unity. The horizontal-slice representation of a general T2 FS (Sect. 6.7.3) uses slices in the third dimension, which exist for closed IT2 FSs but not for IVFSs.

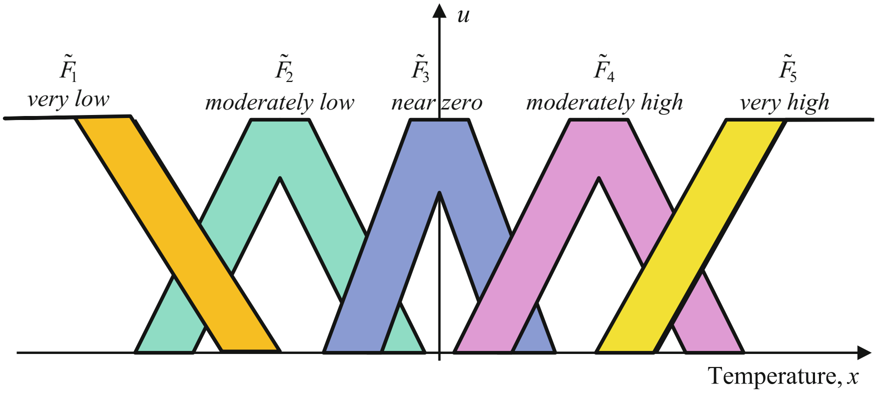
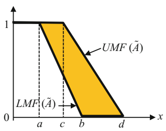
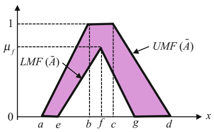
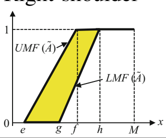


Fig. 6.13 FOUs for Fig. 6.1 MFs

Example 6.11 An example of an FOU for a CIT2 FS is depicted in Fig. 6.12a by using shaded regions. The uniform shading indicates that the FOU is for a CIT2 FS. Shown also on Fig. 6.12a are the supports of two secondary MFs and in Fig. 6.12b the two secondary MFs at x_1 and x_2.

Example 6.12 Figure 6.13 is an "uncertain" version of Fig. 6.1, in which uncertainty has been assumed about the knowledge of where to locate the triangle apex and basepoints, and where to locate the shoulder points of the two end MFs. As in Fig. 6.11a, the uniformly shaded FOUs denote the fact that it is for a CIT2 FS.

Table 6.1 Formulas for upper and lower MFs for piecewise linear left shoulder, interior and right shoulder FOUs

FOU	$\overline{\mu}_{\tilde{A}}(x)$	$\underline{\mu}_{\tilde{A}}(x)$
Left shoulder 	$\overline{\mu}_{\tilde{A}}(x) = \begin{cases} 1 & 0 \le x \le c \\ \frac{d-x}{d-c} & c < x \le d \\ 0 & x > d \end{cases}$	$\underline{\mu}_{\tilde{A}}(x) = \begin{cases} 1 & 0 \le x \le a \\ \frac{b-x}{b-a} & a < x \le b \\ 0 & x > b \end{cases}$
Interior 	$\overline{\mu}_{\tilde{A}}(x) = \begin{cases} 0 & x \le a \\ \frac{x-a}{b-a} & a < x \le b \\ 1 & b < x \le c \\ \frac{d-x}{d-c} & c < x \le d \\ 0 & x > d \end{cases}$	$\underline{\mu}_{\tilde{A}}(x) = \begin{cases} 0 & x \le e \\ \frac{x-e}{f-e}\mu_f & e < x \le f \\ \frac{g-x}{g-f}\mu_f & f < x \le g \\ 0 & x > g \end{cases}$
Right shoulder 	$\overline{\mu}_{\tilde{A}}(x) = \begin{cases} 0 & x \le e \\ \frac{x-e}{f-e} & e < x \le f \\ 1 & f < x \le M \end{cases}$	$\underline{\mu}_{\tilde{A}}(x) = \begin{cases} 0 & x \le g \\ \frac{x-g}{h-g} & g < x \le h \\ 1 & h < x \le M \end{cases}$

Formulas for the upper and lower MFs for piecewise linear left shoulder, interior and right shoulder FOUs are given in Table 6.1.

Example 6.13 As another example of an FOU, the word survey described in Sect. 4.4.2 is returned to. Figure 6.14 indicates how the uncertainties associated with the two endpoints of an interval can be translated into an FOU for pre-chosen triangular type-1 MFs. Let $x = m_a$ denote the average value for the left-hand point of subject data intervals and $x = m_b$ denote the average value for the right-hand point of subject data intervals. The standard deviation for the location of the left-hand point is denoted σ_a, and the standard deviation for the location of the right-hand point is denoted σ_b. m_a and m_b are shown as solid circles in Fig. 6.14.

Uncertainty intervals are defined for m_a and m_b, as $[m_a - \sigma_a, m_a + \sigma_a]$ and $[m_b - \sigma_b, m_b + \sigma_b]$, respectively. There are two cases that must be considered: (1) $m_b - \sigma_b > m_a + \sigma_a$ and (2) $m_b - \sigma_b < m_a + \sigma_a$. In the first case (Fig. 6.14), the uncertainty interval for m_a does not overlap with the uncertainty interval for m_b, but in the second case it does. A construction for the FOU is provided next for the first case, and the construction of the FOU for the second case is left as an exercise (Exercise 6.2).

When $m_b - \sigma_b > m_a + \sigma_a$, as in Fig. 6.14:

1. Let $m_b - m_a = l$.
2. Locate the apex of the triangle at $l/2$, and assign it unity height.
3. The left-hand vertex of the triangle, on the horizontal axis, can range from $m_a - \sigma_a$ to $m_a + \sigma_a$; the region of uncertainty for the left-hand leg is the shaded left-hand triangle whose vertices are at: $(m_a - \sigma_a, 0)$, $(m_a + \sigma_a, 0)$ and $(l/2, 1)$.
4. The right-hand vertex of the triangle, on the horizontal axis, can range from $m_b - \sigma_b$ to $m_b + \sigma_b$; the region of uncertainty for the right-hand leg is the shaded right-hand triangle whose vertices are at: $(l/2, 1)$, $(m_b - \sigma_b, 0)$ and $(m_b + \sigma_b, 0)$.
5. The FOU is the union of all points in the two shaded triangles. This construction also gives some meaning to the FOU shown in Fig. 6.12a.

This is by no means the only way in which an FOU can be associated with the two endpoints of an interval for a linguistic term. Section 10.4 describes another (better) choice.

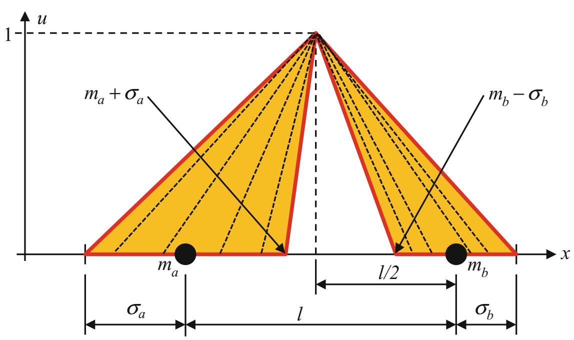


Fig. 6.14 FOU when interval end-point information is requested. Uncertainty interval for m_a does not overlap with the uncertainty interval for m_b. The *dashed black lines* are representative type-1 fuzzy set MFs for each subject

Example 6.14 Ulu et al. (2013) define a *rectangular type-2 fuzzy granule* as a rectangular prism whose base area represents the FOU and height $\alpha \in [0,1]$ represents the secondary MFs. When $\alpha = 1$ this fuzzy granule becomes a *rectangular interval type-2 fuzzy granule*. A T2 FS whose FOU is formed by using rectangular interval type-2 fuzzy granules is called a *Granular type-2 fuzzy set*, and a CIT2 FS whose FOU is formed by using rectangular interval type-2 fuzzy granules is called a *Granular CIT2 FS*.

Each rectangular interval type-2 fuzzy granule is characterized by four parameters (Fig. 6.15a) x_L, x_R, u_L and u_R; however, because adjacent granules touch, a Granular CIT2 FS comprised of K granules is characterized by $4 + 3(K-1)$ parameters. Ulu et al. (2013) note that Granular type-2 MFs provide an "opportunity to express the [MF] uncertainties without any dependence on the shape of a specific function." Unfortunately, this shape independence comes at the expense of many more parameters than are needed to describe a nongranular CIT2 FS, e.g., the FOU in Fig. 6.15b can be approximated by using trapezoidal lower and upper MFs, with a total of 9 parameters (this assumes that the UMF is a normal type-1 fuzzy set, so it can be defined by 4 parameters, and that the LMF is subnormal so it can be defined by 5 parameters, one of which includes a parameter for its height), whereas the four granules in that figure need 13 parameters to describe them. Additionally, because of the discontinuous nature of adjacent granules, K must also be considered as a design parameter.

Because of the non-parsimonious nature of Granular CIT2 FS they are not used in the rest of this book.

Definition 6.20 A *primary MF* (Aisbett et al. 2010) is a member of a family of functions into the unit interval which are parameterized by Ω, i.e., $\mu_\varphi : X \to [0,1]$, $\varphi \in \Omega$. For short, $\mu_A(x)$ is used to denote a primary MF. It will be subject to some restrictions on its parameters. The family of all primary MFs creates an FOU for a CIT2 FS.

Example 6.15 An example of a primary MF is the triangle, depicted in Fig. 6.3a whose vertices (parameters) have been assumed to vary over some interval of values. The FOU associated with this primary MF is shown in Fig. 6.3c.

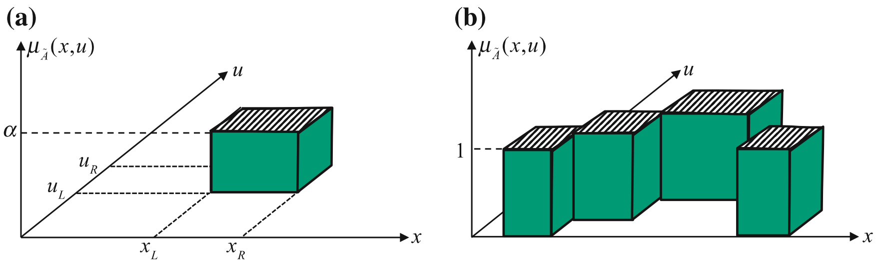


Fig. 6.15 **a** A rectangular type-2 fuzzy granule, and **b** an example of a Granular IT2 FS. In **b** when the *top shaded rectangles* are moved down to the $x-u$ plane, the result is the FOU of $\tilde{A}$

Definition 6.21 (Garibaldi et al. 2005) A *nonstationary fuzzy set* $\dot{A}$ is characterized by a MF $\mu_{\dot{A}}(x,t)$, where $x \in X$, $\mu_{\dot{A}}(x,t) \in [0,1]$ and t is a free variable, *time*—the time at which the fuzzy set is instantiated, i.e., in fuzzy set notation:

$$\dot{A} = \int_{x \in X} \mu_{\dot{A}}(x,t)/x \tag{6.21}$$

Beginning with a type-1 MF, its three main alternative kinds of non-stationarity are: variation in location, variation in slope and noise variation. For example, let c denote the center value of a type-1 MF and $c(t) \equiv c + kf(t)$ where $f(t)$ is called a "perturbation function" which may be random, hence the terminology "nonstationary fuzzy set".

When $f(t)$ is a known deterministic function, then $\mu_{\dot{A}}(x,t)$ can be lower and upper bounded, in which case there is a direct connection between $\dot{A}$ and a CIT2 FS. On the other hand, when $f(t)$ is random (a random process), then $\dot{A}$ is a *random fuzzy set* (Lushu 1995). Such fuzzy sets are very different from fuzzy random variables (e.g., Buckley 2003; Moller and Beere 2004), and can be treated as nonlinear transformations of random processes. If the distribution function can be computed for the now random $\mu_{\dot{A}}(x,t)$, then perhaps lower and upper probability bounds can be established for each value of a primary variable. These bounds might then be somehow related to the lower and upper MFs of a CIT2 FS. How to carry out such calculations remains to be explored.

Nonstationary fuzzy sets are not used in this book.

6.4 Examples of Two Popular FOUs

Two very popular FOUs are for a Gaussian primary MF with uncertain standard deviation and a Gaussian primary MF with uncertain mean. This section provides lots of information about both of these FOUs. Exercise 6.6 asks the reader to provide the comparable information for Gaussian primary MF with uncertain mean and standard deviation. The popularity of these FOUs is due to their parsimony and differentiability. The latter is important when a derivative-based optimization algorithm is used to optimize MF parameters during the design of an interval type-2 fuzzy system (see Sect. 10.2.3).

Example 6.16 **Gaussian primary MF with uncertain standard deviation**

Consider the case of a Gaussian primary MF having a fixed mean, m, and an uncertain standard deviation that takes on values in $[\sigma_1, \sigma_2]$, i.e.,

$$\mu_A(x) = \exp\left[-\tfrac{1}{2}((x-m)/\sigma)^2\right] \quad \sigma \in [\sigma_1, \sigma_2] \tag{6.22}$$

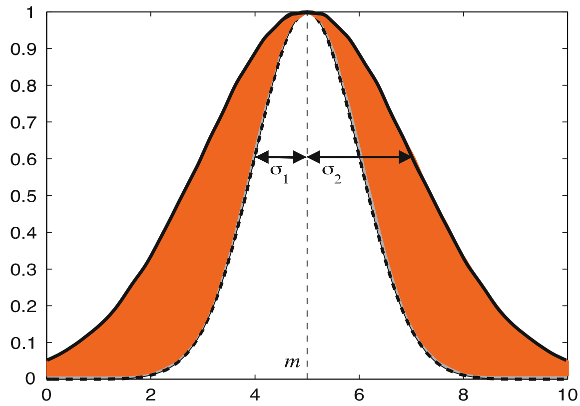


Fig. 6.16 FOU for Gaussian primary MF with uncertain standard deviation

Corresponding to each value of σ, a different membership curve is obtained. The uniform shading for the FOU (Fig. 6.16) again indicates that it is for a CIT2 FS. This primary MF and its associated interval type-2 MFs are used in Sect. 10.3.2 to model measurements that have been corrupted by non-stationary additive noise, and in Sect. 10.6 for rule-based classification of video traffic.

The thick solid curve in Fig. 6.16 denotes the UMF, and the thick dashed curve denotes the LMF; they can be expressed as:

$$\overline{\mu}_{\tilde{A}}(x) = N(x; m, \sigma_2) \tag{6.23}$$

$$\underline{\mu}_{\tilde{A}}(x) = N(x; m, \sigma_1), \tag{6.24}$$

where for example, $N(x; m, \sigma_1) \equiv \exp[-\frac{1}{2}((x-m)/\sigma_1)^2]$. Note that both the upper and lower MFs do not change formulas over $x \in X$ and they are differentiable over $x \in X$, i.e.,

$$\frac{\partial \overline{\mu}_{\tilde{A}}(x)}{\partial m} = \partial N(x; m, \sigma_2)/\partial m = (x-m)N(x; m, \sigma_2)/\sigma_2^2 \tag{6.25}$$

$$\frac{\partial \overline{\mu}_{\tilde{A}}(x)}{\partial \sigma_2} = \partial N(x; m, \sigma_2)/\partial \sigma_2 = (x-m)^2 N(x; m, \sigma_2)/\sigma_2^3 \tag{6.26}$$

$$\frac{\partial \underline{\mu}_{\tilde{A}}(x)}{\partial m} = \partial N(x; m, \sigma_1)/\partial m = (x-m)N(x; m, \sigma_1)/\sigma_1^2 \tag{6.27}$$

$$\frac{\partial \underline{\mu}_{\tilde{A}}(x)}{\partial \sigma_1} = \partial N(x; m, \sigma_1)/\partial \sigma_1 = (x-m)^2 N(x; m, \sigma_1)/\sigma_1^3 \tag{6.28}$$

As will be seen in Example 6.17, the derivatives in (6.25)–(6.28) are much simpler to compute than those for a Gaussian primary MF with uncertain mean; however, this fact should not be the deciding point as to which kind of an FOU to

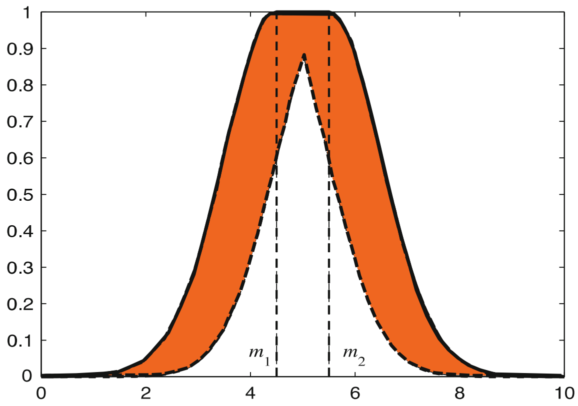


Fig. 6.17 FOU for Gaussian primary MF with uncertain mean

use in a specific application. An understanding about the nature of the application's uncertainties should always be the driving force behind this decision.

Example 6.17 **Gaussian primary MF with uncertain mean**

Consider the case of a Gaussian primary MF having a fixed standard deviation, σ, and an uncertain mean that takes on values in $[m_1, m_2]$, i.e.,

$$\mu_A(x) = \exp\left[-\frac{1}{2}((x-m)/\sigma)^2\right] \quad m \in [m_1, m_2] \tag{6.29}$$

Corresponding to each value of m a different membership curve is obtained. The uniform shading for the FOU (Fig. 6.17) again indicates that it is for a CIT2 FS. This primary MF and its associated interval type-2 MF are used in Chap. 10 for forecasting of time series.

As in Fig. 6.16, the thick solid curve in Fig. 6.17 denotes the UMF, and the thick dashed curve denotes the LMF; they can be expressed as:

$$\overline{\mu}_{\tilde{A}}(x) = \begin{cases} N(x; m_1, \sigma) & x < m_1 \\ 1 & m_1 \le x \le m_2 \\ N(x; m_2, \sigma) & x > m_2 \end{cases} \tag{6.30}$$

$$\underline{\mu}_{\tilde{A}}(x) = \begin{cases} N(x; m_2, \sigma) & x \le \frac{m_1 + m_2}{2} \\ N(x; m_1, \sigma) & x > \frac{m_1 + m_2}{2} \end{cases} \tag{6.31}$$

Note that both the upper and lower MFs change formulas over $x \in X$, but they are differentiable over $x \in X$, i.e.,

$$\frac{\partial \overline{\mu}_{\tilde{A}}(x)}{\partial m_1} = \begin{cases} \partial N(x; m_1, \sigma)/\partial m_1 \\ 0 \\ 0 \end{cases} = \begin{cases} (x - m_1)N(x; m_1, \sigma)/\sigma^2 & x < m_1 \\ 0 & m_1 \le x \le m_2 \\ 0 & x > m_2 \end{cases} \tag{6.32}$$

$$\frac{\partial \overline{\mu}_{\tilde{A}}(x)}{\partial m_2} = \begin{cases} 0 \\ 0 \\ \partial N(x; m_2, \sigma)/\partial m_2 \end{cases} = \begin{cases} 0 & x \le m_1 \\ 0 & m_1 \le x \le m_2 \\ (x - m_2)N(x; m_2, \sigma)/\sigma^2 & x \ge m_2 \end{cases} \tag{6.33}$$

$$\frac{\partial \overline{\mu}_{\tilde{A}}(x)}{\partial \sigma} = \begin{cases} \partial N(x; m_1, \sigma)/\partial \sigma \\ 0 \\ \partial N(x; m_2, \sigma)/\partial \sigma \end{cases} = \begin{cases} (x - m_1)^2 N(x; m_1, \sigma)/\sigma^3 & x < m_1 \\ 0 & m_1 \le x \le m_2 \\ (x - m_2)^2 N(x; m_2, \sigma)/\sigma^3 & x > m_2 \end{cases} \tag{6.34}$$

$$\frac{\partial \underline{\mu}_{\tilde{A}}(x)}{\partial m_1} = \begin{cases} 0 \\ \partial N(x; m_1, \sigma)/\partial m_1 \end{cases} = \begin{cases} 0 & x \le \dfrac{m_1 + m_2}{2} \\ (x - m_1)N(x; m_1, \sigma)/\sigma^2 & x > \dfrac{m_1 + m_2}{2} \end{cases} \tag{6.35}$$

$$\frac{\partial \underline{\mu}_{\tilde{A}}(x)}{\partial m_2} = \begin{cases} \partial N(x; m_2, \sigma)/\partial m_2 \\ 0 \end{cases} = \begin{cases} (x - m_2)N(x; m_2, \sigma)/\sigma^2 & x \le \dfrac{m_1 + m_2}{2} \\ 0 & x > \dfrac{m_1 + m_2}{2} \end{cases} \tag{6.36}$$

$$\frac{\partial \underline{\mu}_{\tilde{A}}(x)}{\partial \sigma} = \begin{cases} \partial N(x; m_2, \sigma)/\partial \sigma \\ \partial N(x; m_1, \sigma)/\partial \sigma \end{cases} = \begin{cases} (x - m_2)^2 N(x; m_2, \sigma)/\sigma^3 & x \le \dfrac{m_1 + m_2}{2} \\ (x - m_1)^2 N(x; m_1, \sigma)/\sigma^3 & x > \dfrac{m_1 + m_2}{2} \end{cases} \tag{6.37}$$

The important points to remember from these calculations are that, for a Gaussian primary MF with uncertain mean, one must be very careful to use the correct values for the derivatives of the upper and lower MFs with respect to their parameters, and these derivatives depend on where the independent variable x is in relation to the means of the left- and right-hand Gaussians, i.e., they depend on which branch of a lower or upper MF is *active*.[16]

6.5 Interval Type-2 Fuzzy Numbers

What exactly is an interval type-2 fuzzy number? There does not seem to be general agreement about the answer to this question, unlike the general agreement that exists about what a type-1 fuzzy number is (see Definition 2.5). This is because an IT2 FS is described by two T1 FSs, its lower and upper MFs, and so different possibilities avail themselves for an IT2 FS to be called an interval type-2 fuzzy number. Hamrawi and Coupland (2009) define two kinds of interval type-2 fuzzy numbers[17]:

---

[16]The concept of an *active branch* was first described in Liang and Mendel (2000).

[17]Interestingly, Hamrawi and Coupland (2009) also define different kinds of general type-2 fuzzy numbers. See Sect. 11.3.

- A *perfectly normal interval type-2 fuzzy number* is an IT2 FS both of whose lower and upper MFs of its FOU are type-1 fuzzy numbers.
- A *normal interval type-2 fuzzy number* is an IT2 FS only whose upper MF of its FOU is a type-1 fuzzy number.

I will now argue why, to me, only a "perfectly normal interval type-2 fuzzy number" deserves the designation of an interval type-2 fuzzy number. To begin, anything that is called a "fuzzy number" should be a model for a linguistic term that is related to the number, e.g., if n is a real number, then a fuzzy number would be for terms like *close to n*, *very close to n*, *around n*, *about n*, etc. So, at n there is no uncertainty (n is n), which means that at n the membership grade for the fuzzy number should be 1 (as occurs for a perfectly normal interval type-2 fuzzy number) and not an interval of values (as occurs for a normal interval type-2 fuzzy number).

Continuing, suppose that a group of S subjects is asked: "On a scale of l to r where would you locate the endpoints (a and b) of an interval of real numbers for a linguistic term related to the real number n (of course, $l \leq n \leq r$), e.g., *around n*. The result will be a set of S intervals, $[a^{(i)}, b^{(i)}]_{i=1}^{S}$. After doing some preprocessing of these S intervals (e.g., remove outliers, keep only the intervals that contain n, keep only the intervals that overlap,[18] etc.), one will be left with S' intervals, $[a'^{(i)}, b'^{(i)}]_{i=1}^{S'}$, where $S' \leq S$. Overlaying these S' intervals, one will find a common interval, $[c, d]$, where $[c, d] \subset [a'^{(i)}, b'^{(i)}]_{i=1}^{S'}$. The phrase "common interval" means all S' subjects are in agreement about $[c, d]$; hence, there is no uncertainty about $[c, d]$. Consequently, the membership grade should be 1 for $[c, d]$. When an IT2 FS is used to model this fuzzy number, it is only a perfectly normal interval type-2 fuzzy number that can achieve this.

By these arguments, only a "perfectly normal interval type-2 fuzzy number" deserves the designation of an interval type-2 fuzzy number. Although, not everyone may agree with this, then, at the very least, one should use the above Hamrawi–Coupland designations. In this book the following will be abided by:

Definition 6.22 An *interval type-2 fuzzy number* is an IT2 FS whose lower and upper MFs of its FOU are type-1 fuzzy numbers.

Consequently, the Example 6.16 Gaussian primary MF with uncertain standard deviation is an interval type-2 fuzzy number, but the Example 6.17 Gaussian primary MF with uncertain mean is not.

Interval type-2 fuzzy numbers will play an important role in one kind of non-singleton fuzzification in an IT2 fuzzy system (see Sect. 9.4.2.3).

[18]Not only is it true that "Words mean different things to different people," but, in addition, it is true that "Words must mean similar things to different people," or else people are not communicating effectively (Liu and Mendel 2008). The former adage is the motivation for modeling linguistic terms (words) using type-2 fuzzy sets, whereas the latter adage is the motivation for keeping only overlapping intervals.

6.6 Different Kinds of T2 FSs: Hierarchy

As noted in Mendel (2014, footnote 2): "In the early days of type-2 fuzzy sets and systems, the phrases 'type-2 fuzzy set' or 'type-2 fuzzy system' were used in an all-inclusive way, meaning any kind of type-2 fuzzy set or system. During the past 15 years [now more than 17 years] most of the attention has been given to [closed] interval type-2 fuzzy sets and systems. It is only within the past 5 years or so that there has been a return to more general type-2 fuzzy sets and systems, and, to distinguish them from the more specialized [closed] interval type-2 fuzzy sets and systems, the term 'general' is being used. In essence, type-2 fuzzy sets and systems now consist of the union of interval and general type-2 fuzzy sets and systems." This is captured in the Venn diagram of Fig. 6.18.

As mentioned just before Definition 6.1, T2 FSs have been beset with some problematic notations (see Mendel et al. 2016) for a comprehensive historical explanation of them, as well as recommended fixes, all of which have been adopted in this book). The most problematic notation was the use of $J_x \subseteq [0, 1]$ in the definition of $\tilde{A}$, i.e., instead of (6.1) $\tilde{A}$ was defined as:

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\}.$$

Additionally, nowhere in Mendel (2001) was J_x defined formulaically, as has been done in (6.4a).

This book emphasizes the use of I_x instead of J_x, where $J_x = \{x\} \times I_x$. All of the early works on IT2 FSs were for when I_x is connected; however, by only stating that $J_x \subseteq [0, 1]$, and not defining J_x, this left $J_x \subseteq [0, 1]$ open to the richer interpretation provided in Bustince et al. (2015) for which I_x does not have to be connected. It is in that interpretation that one gains an appreciation for distinguishing between connected and disconnected I_x.

Example 6.18 In Bustince et al. (2015) one finds examples of different kinds of IT2 FSs, one of which is closed, and the others of which are not. Regarding the latter:

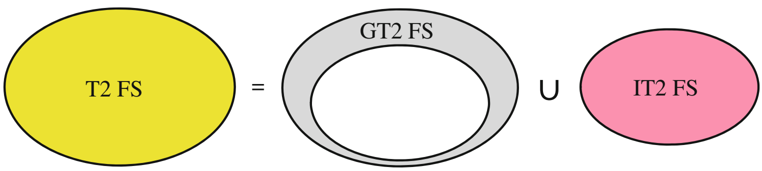


Fig. 6.18 T2 FSs and systems now consist of the union of interval and general T2 FSs and systems

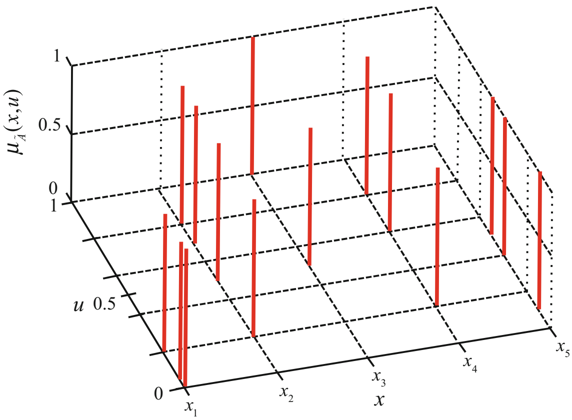


Fig. 6.19 Example of a *multiset* from an IT2 FS (adapted from Bustince et al. 2015)

- Figure 6.19 is an example of a *multiset* from an IT2 FS for which I_{x_i} is a set of points each with a secondary grade of unity. In this example,

$$\mu_{\tilde{A}}(x_1,u) = \begin{cases} 1 & u \in I_{x_1} = \{0, 0.05, 0.2\} \\ 0 & \text{otherwise} \end{cases}, \quad \mu_{\tilde{A}}(x_2,u) = \begin{cases} 1 & u \in I_{x_2} = \{0.2, 0.5, 0.7, 0.8\} \\ 0 & \text{otherwise} \end{cases},$$
$$\mu_{\tilde{A}}(x_3,u) = \begin{cases} 1 & u \in I_{x_3} = \{0.5, 1\} \\ 0 & \text{otherwise} \end{cases}, \quad \mu_{\tilde{A}}(x_4,u) = \begin{cases} 1 & u \in I_{x_4} = \{0.2, 0.6, 0.8\} \\ 0 & \text{otherwise} \end{cases}, \text{ and}$$
$$\mu_{\tilde{A}}(x_5,u) = \begin{cases} 1 & u \in I_{x_5} = \{0.1, 0.4, 0.5\} \\ 0 & \text{otherwise} \end{cases}.$$

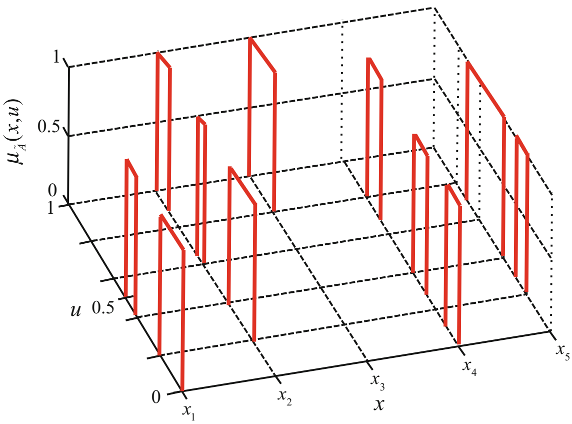


Fig. 6.20 Example of a *multi interval-valued* FS as an IT2 FS (adapted from Bustince et al. 2015)

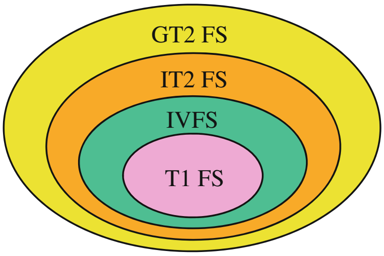


Fig. 6.21 A way to summarize fuzzy sets, in which the ellipses denote "encompass," as is explained in the text (adapted from Bustince et al. 2015)

- Figure 6.20 is an example of a *multi interval-valued* FS as an IT2 FS for which I_{x_i} is a set of nonoverlapping intervals, each with a secondary grade of unity for all of their elements. In this example:

$$\mu_{\tilde{A}}(x_1,u)=\begin{cases}1 & u\in I_{x_1}=[0,0.2]\cup[0.4,0.5]\\ 0 & \text{otherwise}\end{cases},\ \mu_{\tilde{A}}(x_2,u)=\begin{cases}1 & u\in I_{x_2}=[0.2,0.4]\cup[0.6,0.66]\cup[0.9,1]\\ 0 & \text{otherwise}\end{cases}$$
$$\mu_{\tilde{A}}(x_3,u)=\begin{cases}1 & u\in I_{x_3}=[0.8,1]\\ 0 & \text{otherwise}\end{cases},\ \mu_{\tilde{A}}(x_4,u)=\begin{cases}1 & u\in I_{x_4}=[0,0.1]\cup[0.3,0.4]\cup[0.7,0.8]\\ 0 & \text{otherwise}\end{cases},\ \text{and}$$
$$\mu_{\tilde{A}}(x_5,u)=\begin{cases}1 & u\in I_{x_5}=[0.2,0.3]\cup[0.4,0.7]\\ 0 & \text{otherwise}\end{cases}.$$

Of course, one can combine these two examples so that I_{x_i} is a set of points and nonoverlapping intervals, each with a secondary grade of unity for all of their elements. An example of doing this can also be found in Bustince et al. (2015), and is also left as an exercise for the reader (Exercise 6.10).

Figure 6.21 is adapted from Fig. 8 in Bustince et al. (2015). It should be interpreted in the following way: When a T1 FS and an IVFS are expressed as T2 FSs (see Sect. 6.8 for how to do this), then a GT2 FS encompasses an IT2 FS which encompasses an IVFS[19] which encompasses a T1 FS. This seems to be a very nice way to summarize the different kinds of fuzzy sets.

The rest of this book focuses for the most part only on closed general[20] *and closed IT2 FSs,* because, as of the year 2016, it is only for such T2 FSs that one knows how to perform all of the operations that are needed to implement either an interval or general type-2 fuzzy system, and it is only those fuzzy systems that have been applied to real world applications. Consequently, *in the rest of this book when the phrase "interval (or general) T2 FS (or system)" is used it is always to be understood (unless indicated otherwise) that this means a "closed interval (or general) T2 FS (or system)."* The abbreviations *IT2 FS* and *GT2 FS* that are used in the rest of this book are always for such type-2 fuzzy sets.

[19]The elevated IVFS will then be the same as a CIT2 FS.

[20]A *closed GT2 FS* is defined in Definition 6.25.

6.7 Mathematical Representations for T2 FSs[21]

There are four ways to mathematically represent a T2 FS[22] (Mendel 2014): (a) collection of points as in (6.1); (b) union of vertical slices (over $x \in X$), where each vertical slice (see Fig. 6.5) is a T1 FS (a secondary MF); (c) union of wavy slices, where each wavy slice is an embedded T2 FS (see Fig. 6.10); and, (4) fuzzy union of horizontal slices (over $\alpha \in [0, 1]$), where each horizontal slice resembles an IT2 FS raised to level α.

The *collection of points representation* is the starting point for all of the other representations, and often corresponds to the way that data about a GT2 FS can be stored; but, it does not seem to be good for much else, so nothing else will be said about it.

Just as it is possible to rigorously prove that a type-1 MF can be represented as the fuzzy union of all of its α-cuts (each raised to level α), it is also possible to rigorously prove that a type-2 MF can be represented as the union of all of its vertical slices, or horizontal slices or wavy slices. In retrospect, all of these decompositions are visually obvious.

6.7.1 Vertical Slice Representation

Theorem 6.1 *The* vertical-slice representation *of GT2 FS $\tilde{A}$ focuses on each value of the primary variable x, and expresses* (6.1) *as the union of all of its secondary T1 FSs, i.e.,*

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}(x)}(u)/x \tag{6.38}$$

where

$$\mu_{\tilde{A}(x)}(u) = \int_{u \in [0,1]} f_x(u)/u \tag{6.39}$$

so that[23]

[21]Some of the material in this section is adapted from Mendel (2014) modulo some notational changes.

[22]Be reminded that in this book all secondary MFs are convex T1 FSs.

[23]The first statement of the vertical-slice representation in terms of its α-cuts is Mendel (2014).

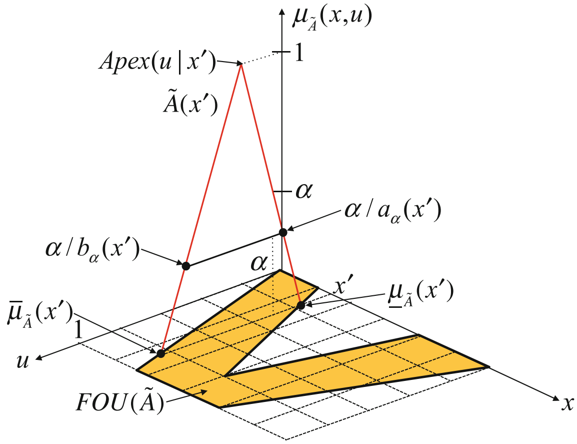


Fig. 6.22 A secondary set $\tilde{A}(x')$, its triangular MF, and an α-cut of $\tilde{A}(x')$ raised to level α. Observe that $\tilde{A}(x')$ is anchored on FOU($\tilde{A}$) at $\underline{\mu}_{\tilde{A}}(x')$ and $\overline{\mu}_{\tilde{A}}(x')$

$$\tilde{A} = \int_{x\in X}\left[\bigcup_{\alpha\in[0,1]}\left[\alpha/\tilde{A}(x)_\alpha\right]\right]\Big/x = \int_{x\in X}\sup_{\alpha\in[0,1]}\left[\alpha/\tilde{A}(x)_\alpha\right] \tag{6.40}$$

in which the α*-cut of the T1 FS* $\tilde{A}(x)$, $\tilde{A}(x)_\alpha$, *is given by (see* Definition 2.9, Example 2.27 *and* Fig. 6.22*)*

$$\tilde{A}(x)_\alpha = \{u|\mu_{\tilde{A}(x)}(u)\geq\alpha\} \equiv [a_\alpha(x), b_\alpha(x)] \tag{6.41}$$

Proof Equation (6.39) is already stated in (6.3), and (6.38) collects all of the vertical slices. At each $x\in X$, $\tilde{A}(x)$ has the following $\alpha-$cut decomposition for convex secondary MFs (Theorem 2.3):

$$\begin{aligned}\tilde{A}(x) &= \bigcup_{\alpha\in[0,1]}\left[\alpha/\tilde{A}(x)_\alpha\right] = \sup_{\alpha\in[0,1]}\left[\alpha/\tilde{A}(x)_\alpha\right]\\ &= \sup_{\alpha\in[0,1]}\left[\alpha/[a_\alpha(x), b_\alpha(x)]\right]\end{aligned} \tag{6.42}$$

The first line of (6.42) is the right-hand portion of (6.40), and the second line of (6.42) uses $\tilde{A}(x)_\alpha$ in (6.41), which completes the proof of Theorem 6.1.

The vertical-slice representation of $\tilde{A}$ in (6.38) and (6.39) was used originally in Mizumoto and Tanaka (1976, 1981) to derive formulas for set-theoretic operations of GT2 FSs. It is excellent for computations involving GT2 FSs and can also be used for theoretical studies. In fact, as is explained in Sects. 6.7.4 and 11.14, it is one of the preferred representations for solving real-world general type-2 fuzzy system optimal design problems. The vertical-slice representation of $\tilde{A}$ in (6.40) strongly suggests that α-cuts will play a useful and central role for GT2 FSs. That this is so will be made very clear in Chaps. 7 and 11.

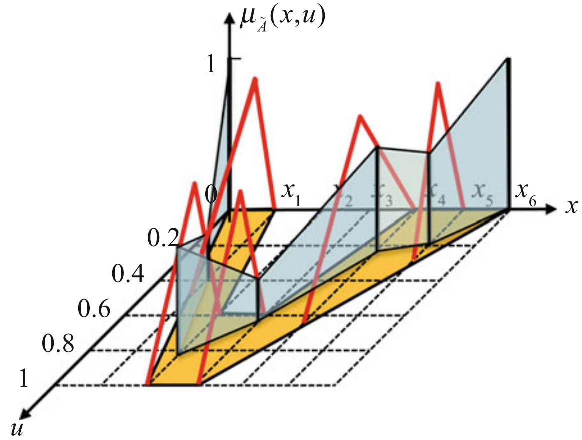


Fig. 6.23 A wavy slice for the Fig. 6.9 GT2 FS. Each secondary MF contributes one element to the wavy slice. The shading of the wavy slice is only for artistic purposes; however, for continuous X and U the shaded region becomes a continuous foil

6.7.2 *Wavy Slice Representation*

Theorem 6.2 [Wavy Slice Representation Theorem[24] *for a GT2 FS* $\tilde{A}$ (Mendel and John 2002)] *GT2 FS* $\tilde{A}$ *is the set-theoretic union of all of its embedded T2 FSs, i.e.*

$$\tilde{A} = \bigcup_{\forall j} \tilde{A}_e^j \tag{6.43}$$

Proof Equation (6.43) is obvious by using the following simple informal geometric argument. Create all of the possible embedded T2 FSs for $\tilde{A}$ (see Fig. 6.23 for one such set) and take their union to reconstruct $\mu_{\tilde{A}}(x,u)$. Same points, which occur in different embedded T2 FSs only appear once in the set-theoretic union.

Although impractical for most computations, because (6.43) requires the enumeration of all of the embedded T2 FSs, and their number can be enormously large depending upon the discretization levels of the primary and secondary variables, this wavy slice representation of a GT2 FS has proved to be of great value for developing new theoretical results. Arguably, its most important use has been for IT2 FSs, as is encapsulated in the following:

Theorem 6.3 [Wavy Slice Representation Theorem *for an IT2 FS* (Mendel and Wu 2010, Chap. 2)] *The FOU of an IT2 FS is (covered by) the union of all of its embedded T1 FSs, i.e.*[25]

---

[24]This representation theorem, which was originally stated and proved in Mendel and John (2002) for $u \in I_x\,[J_x]$ instead of for $u \in [0,1]$, has also been referred to as the *Mendel-John Representation Theorem*. I prefer "wavy slice" because it fits in very nicely with the description of the other "slice" representations of a GT2 FS. Its extension to non-closed (disconnected) GT2 FSs remains to be explored.

[25]The extension of Theorem 6.3 to non-closed (disconnected) IT2 FSs remains to be explored.

$$\mathrm{FOU}(\tilde{A}) = \bigcup_{\forall j} A_e^j \tag{6.44}$$

Proof Create all of the possible embedded T1 FSs in $\mathrm{FOU}(\tilde{A})$ and take their union to reconstruct $\mathrm{FOU}(\tilde{A})$. Same points, which occur in different embedded T1 FSs, only appear once in the set-theoretic union.

Note that Theorem 6.3 can be interpreted as a *covering representation*, because the union of all embedded T1 FSs covers the entire FOU.

Definition 6.23 A *maximal covering* of $\mathrm{FOU}(\tilde{A})$ is one in which every possible embedded T1 FS is used to cover the FOU, and so same points will occur in (many) different embedded T1 FSs. A *minimal covering* of $\mathrm{FOU}(\tilde{A})$ is one that uses a minimal number of embedded T1 FSs to cover the FOU.

Example 6.19 A *minimal covering* of $\mathrm{FOU}(\tilde{A})$ can be achieved by choosing embedded T1 FSs as a linear combination of the lower and upper MFs of $\tilde{A}$ as follows:

$$A_e^j = (1-j)\mathrm{LMF}(\tilde{A}) + j\mathrm{UMF}(\tilde{A}), \quad j \in [0,1] \tag{6.45}$$

This is also called a *homotopy*.[26]

When, for example, (6.45) is applied to the FOU in Fig. 6.16, then that FOU will be covered by Gaussian type-1 fuzzy sets each of which passes through the common point $(m, 1)$. See, also Exercises 6.13 and 6.14.

By using Theorem 6.3 it is possible to apply everything that has already been developed for T1 FSs to each of the embedded T1 FSs, after which the union in (6.44) is performed. This has been done in Mendel et al. (2006) and will be demonstrated in Chap. 7 (Examples 7.15–7.17). Additionally, Mendel (2009) has demonstrated that Theorem 6.3 is a very good place to begin when trying to solve a new problem that involves IT2 FSs.

6.7.3 *Horizontal Slice Representation*[27]

Definition 6.24 An *α-plane* (Liu 2008; Mendel et al. 2009) for a GT2 FS $\tilde{A}$, denoted $\tilde{A}_\alpha$, is the union of all primary memberships of $\tilde{A}$ whose secondary grades are greater than or equal to $\alpha \in [0,1]$, i.e.

---

[26]A *homotopy* between two continuous functions f and g from a topological space X to a topological space Y is defined to be a continuous function $H : X \times [0,1] \to Y$ from the product of the space X with the unit interval $[0,1]$ to Y such that, if $x \in X$ then $H(x,0) = f(x)$ and $H(x,1) = g(x)$.

[27]If a reader is interested only in IT2 FSs and fuzzy systems and not GT2 FSs and systems, this section as well as Sect. 6.7.4 can be omitted.

$$\begin{aligned}\tilde{A}_\alpha &= \left\{(x,u), \mu_{\tilde{A}}(x,u) \geq \alpha | x \in X, u \in [0,1]\right\} \\ &= \int_{x \in X} \int_{u \in [0,1]} \{(x,u) | f_x(u) \geq \alpha\}\end{aligned} \tag{6.46}$$

Alternatively, $\tilde{A}_\alpha$ can be expressed by means of (6.41), as:

$$\tilde{A}_\alpha = \int_{x \in X} \tilde{A}(x)_\alpha / x = \int_{x \in X} [a_\alpha(x), b_\alpha(x)]/x \tag{6.47}$$

$\tilde{A}_\alpha$ has a LMF and an UMF, where $(x \in X)$:

$$\begin{cases} \mathrm{LMF}(\tilde{A}_\alpha) = a_\alpha(x) \\ \mathrm{UMF}(\tilde{A}_\alpha) = b_\alpha(x) \end{cases} \tag{6.48}$$

When $\tilde{A}_\alpha$ is raised to level- α, it is a plane (horizontal slice) at that level, and can be obtained by connecting respective α-cuts of all type-1 vertical slice secondary MFs for $x \in X$.

Theorem 6.4 *The* horizontal-slice representation *of GT2 FS* $\tilde{A}$ *is*:

$$\tilde{A} = \sup_{\alpha \in [0,1]} \alpha / \left[\int_{x \in X} [a_\alpha(x), b_\alpha(x)]/x\right] = \sup_{\alpha \in [0,1]} \alpha / \tilde{A}_\alpha = \bigcup_{\alpha \in [0,1]} \alpha / \tilde{A}_\alpha \tag{6.49}$$

Proof This follows directly from substituting (6.41) into the right-hand side of (6.40), and then focusing first on x and then on α, after which (6.47) is used.

An α-plane can be obtained directly from the α-cuts of the secondary MFs. Just as an α-cut of a T1 FS resides on its 1-D domain X, $\tilde{A}_\alpha$ resides on its 2-D domain $X \times U$. Consequently, for a GT2 FS whose secondary MFs are convex T1 FSs each α-plane is an IVFS, and

$$\mathrm{FOU}(\tilde{A}) = \tilde{A}_0. \tag{6.50}$$

Recall (Sect. 2.12) that when an α-cut is raised to level α one obtains an α-level. Analogously, when an α-plane is raised to level α one obtains a *horizontal slice at level* α, $R_{\tilde{A}_\alpha}$ (Mendel 2010), i.e.

$$R_{\tilde{A}_\alpha} = \alpha / \tilde{A}_\alpha \tag{6.51}$$

Table 6.2 Comparisons of α-plane and *z*Slice descriptions (adapted from Mendel 2014)

Item	α-plane description	*z*Slice description
Coordinates	(x, u, μ)	(x, y, z)
α−plane	$\tilde{A}_\alpha = \int_{x\in X}\int_{u\in[0,1]} \{(x,u)\|f_x(u)\geq\alpha\}$	$\tilde{Z}_\alpha$ projected onto $X \times Y$
*z*Slice	$\alpha/\tilde{A}_\alpha = R_{\tilde{A}_\alpha}$	$\tilde{Z}_z = \int_{x\in X}\int_{y\in[0,1]} z/(x,y)$
FOU$(\tilde{A})$	$\tilde{A}_0$	$\tilde{Z}_0$
Vertical slice $(x_j \in X)$	$\int_{\alpha\in[0,1]}\int_{u\in[0,1]} \alpha/(x_j,u)$	$\int_{z\in[0,1]}\int_{y\in[0,1]} z/(x_j,y)$
Representation of $\tilde{A}$	$\tilde{A} = \bigcup_{\alpha\in[0,1]} R_{\tilde{A}_\alpha} = \sup_{\alpha\in[0,1]} [\alpha/\tilde{A}_\alpha]$	$\tilde{A} = \bigcup_{z\in[0,1]} \tilde{Z}_z = \sup_{z\in[0,1]} [\tilde{Z}_z]$

This has been called an "α-plane raised to Level α"or a[28] "zSlice" (Wagner and Hagras 2008, 2010, 2013). Of course, one can also shorten the expression "α-plane raised to Level α" to the much simpler expression "α-level plane." Note, also, $R_{\tilde{A}_\alpha}$ is an IT2 FS all of whose secondary grades equal α (rather than 1 as would be the case for the usual IT2 FS), and that

$$\mathrm{FOU}(R_{\tilde{A}_\alpha}) = \tilde{A}_\alpha \tag{6.52}$$

Equation (6.52) provides an α-plane with an important interpretation of and connection to an IT2 FS of height α.

Example 6.20 Figure 6.24 depicts some α-planes raised to level α for the Fig. 6.9 GT2 FS. It was constructed by fixing a value of α (e.g., $\alpha = 1/3$), drawing the α-cut at level α for each of the secondary MFs, and then connecting the points where those horizontal lines intersect the MFs.[29] Observe that, because all of the secondary MFs

[28]Zadeh (1975) does not use the term α-plane, nor does he have an α-plane decomposition for a T2 FS. The term α-plane was introduced first in Liu (2008), and the term *zSlice* was introduced first in Wagner and Hagras (2008). In the *z*Slice literature coordinates are called (*x, y, z*) instead of (*x, u, μ*); hence, *z* is also the same as α. The connections between these two representations are summarized in Table 6.2.

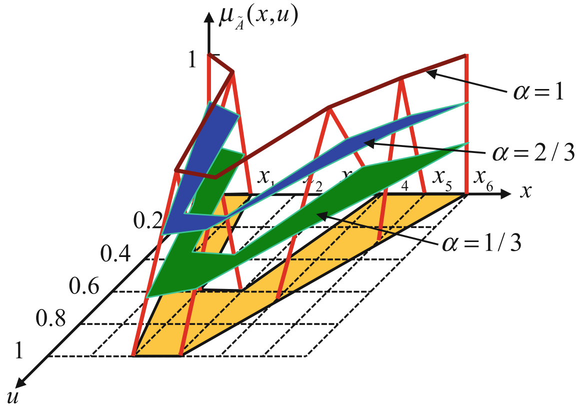
Chen and Kawase (2000) also use α-cuts, and while they do not have a formal representation theorem for a T2 FS, their Definition 3.2 expresses a secondary MF in terms of α-cuts. Tahayori et al. (2006) state a "Representation Principle" without proof; it is the same as Liu's α-plane Representation Theorem, but they do not use the term α-plane.

Starczewski (2009a, b) has done some related work, but he does not have an α-plane Representation Theorem.

Hamrawi et al. (2010) use α-cuts of both the lower and upper MFs of the FOU to represent an IVFS. It also reexamines α-planes and zSlices, and, among other things, shows that a GT2 FS can be represented as the union of all its α-cuts of all its α-planes.

[29]Drawing or sketching 3-D MFs can be quite challenging. See Mendel (2012) for step-by-step instructions for drawing or sketching 2-1/2D MFs, which are 2-D figures that give the appearance of being 3-D.

Fig. 6.24 Some α-planes raised to level α for the Fig. 6.9 GT2 FS



are normal triangles, the $\alpha = 1$ α-plane raised to level α is just a T1 FS, i.e., a function and not a plane.

One can give compelling reasons for using both of the terms "*α-plane*" or "*zSlice*." A reason for using "*α-plane*" is: each type-1 secondary MF can be expressed in terms of alpha-cuts and an alpha-plane is the union of all such alpha-cuts at a fixed value of alpha, so the term "*α-plane*" connects perfectly with the long-used term "α-cut." Two reasons for using "*zSlice*" are: (1) naming coordinates (x, y, z) has a long mathematical heritage, whereas calling them (x, u, μ) does not, and (2) z Slices are already al level z, which is exactly where we want them to be, whereas an α-plane has to be raised to level α for it to be where we want it to be.

It is a bit confusing to have two different terminologies for the same things, so in this book [as in Mendel (2014)] this kind of a representation is called a *horizontal-slice representation* since "horizontal" compliments "vertical" rather nicely. Because my own preference is to connect type-1 and type-2 results smoothly, the *α-plane* terminology is used in the rest of this book.[30]

The great value of the horizontal-slice representation is that it will let everything that is developed for IT2 FSs, in Chaps. 7 and 8, be applied to each of the horizontal slices.

Definition 6.25 A *closed GT2 FS* $\tilde{A}$ is one whose horizontal slices are closed for $\alpha \in [0, 1]$.

This follows from when Definition 6.18 is applied to (6.51) and (6.49) expressed as $\tilde{A} = \bigcup_{\alpha \in [0,1]} R_{\tilde{A}_\alpha}$. The GT2 FS in Fig. 6.24 is closed.

For a GT2 FS to be closed all of its secondary MFs must be convex.

---

[30]If the universes of discourse for primary or secondary variables are not continuous, then it would not be correct to use the word "plane." One could then revert to *horizontal slice at level* α; such a slice would only contain a finite set of discrete x and u, or discrete x and intervals for u. These situations are not considered in this book.

6.7.4 *Which Representations Are Most Useful for Optimal Design[31] Applications?*

Recall, from Chap. 4, that during the optimal design of a type-1 fuzzy system a mathematical objective function, $J(\boldsymbol{\phi})$, was established that depends upon the design parameters, $\boldsymbol{\phi}$. The elements of $\boldsymbol{\phi}$ include all of the antecedent and consequent MF parameters as well as any defuzzification parameters (if there are any). $J(\boldsymbol{\phi})$ is a nonlinear function of $\boldsymbol{\phi}$ and so some sort of mathematical programming approach had to be used to optimize it.

Chapter 10 will demonstrate that a mathematical objective function $J(\boldsymbol{\phi})$ is also established during the optimal designs of an IT2 fuzzy system, and that the design parameters in $\boldsymbol{\phi}$ not only include all of the antecedent and consequent FOU parameters, but also any parameters that are associated with mapping IT2 fuzzy sets into a crisp number.

Chapter 11 will demonstrate that a mathematical objective function $J(\boldsymbol{\phi})$ is also established during the optimal designs of a GT2 fuzzy system, and that the design parameters in $\boldsymbol{\phi}$ not only include all of the antecedent and consequent FOU parameters, but also the secondary MF parameters, and any parameters that are associated with mapping GT2 fuzzy sets into a crisp number.

The basic premise of this book is that *one should use a parsimonious parametric representation of a T2 FS during an optimal design of a T2 fuzzy system, regardless of whether it is uses IT2 FSs or GT2 FSs*. Such a representation is one that is described by as few parameters as possible (Definition 6.14).

Clearly, *a parsimonious representation of an IT2 FS is obtained when its lower and upper MFs are described parsimoniously*, e.g., as in Examples 6.16 and 6.17. Things are not as clear for a GT2 FS.

It is obvious that the point representation of a GT2 FS is not parsimonious, because it is not even a parametric representation. Similarly, the wavy slice representation, in which the wavy slices must be enumerated from all of the points of the GT2 FS, is also a nonparametric representation. Neither of these representations is useful for an optimal design of a GT2 fuzzy system.

The horizontal-slice representation needs to be made parametric for it to be useful for an optimal design of a GT2 fuzzy system. This representation requires choosing the number of horizontal slices and then parameterizing each of them. One way to do this is to assume that all of the horizontal slices have the same shape, and a higher horizontal slice is a *squished* version of the one just below it. A way to squish a horizontal slice at level α_1 into a horizontal-slice at level α_2 $(\alpha_2 > \alpha_1)$ is to begin with the FOU (at level $\alpha_1 = 0$) and then let the UMF (LMF) of the horizontal slice at level α_2 $(\alpha_2 > \alpha_1)$ be a scaled version of the UMF (LMF) of the FOU. The scale factor for the UMF (LMF) has to be less (greater) than unity, and as α increases in values towards its final value of 1, the scale factor has to become

[31] If a reader is interested only in IT2 FSs and fuzzy systems, this section can be omitted.

smaller and smaller (larger and larger), but not so large that the LMF is above the UMF (see Exercise 6.18). By this approach, if there are k horizontal slices and the two scale factors are different, then the number of parameters to represent this kind of GT2 FS is $2k + n_0$, where n_0 denotes the number of parameters that describe the FOU when $\alpha = 0$. If the scale factors are constrained to be the same (see Exercise 6.19), then there will be $k + n_0$ parameters. For a relatively small value of k (2–6) this can be a parsimonious representation.

Yet another way of squishing a horizontal slice at level α_1 into a horizontal-slice at level α_2 $(\alpha_2 > \alpha_1)$ is to begin with the FOU (at level $\alpha_1 = 0$) and use the same UMF of the FOU at all α levels, but let the LMF of the horizontal-slice at level $\alpha_2(\alpha_2 > \alpha_1)$ be a scaled version of the LMF of the FOU (see Exercise 6.20) (Kumbasar and Hagras 2015). By this approach, if there are k horizontal slices, then the number of parameters to represent this kind of GT2 FS is also $k + n_0$.

Our conclusion from this is that *when more than a few horizontal slices are used, the horizontal-slice representation of a GT2 FS may not be a useful representation for the optimal design of a GT2 fuzzy system, but if only a small number of horizontal slices are used, and any of the above squishing techniques are used, then the horizontal-slice representation of a GT2 FS is a useful representation for the optimal design of a GT2 fuzzy system.*[32]

It will now be demonstrated that the vertical-slice representation of a GT2 FS is a very flexible and parsimonious representation of such a fuzzy set.

My suggestion for parameterizing the vertical-slice representation of a GT2 FS is to: (1) Parameterize its FOU exactly as one presently parameterizes the FOU of an IT2 FS; (2) Parameterize the secondary MFs by choosing a fairly simple function that introduces only one new parameter; (3) If performance is not acceptable then use secondary MFs that can be described by using two new parameters; etc. Because the secondary MFs are vertical slices, they are always anchored on the already parameterized FOU. Examples of such secondary MFs are given next.

Example 6.21 **(triangle secondary MFs)**: The vertices of the base of each triangle are located at $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$ (see Fig. 6.22), and the location of the apex of each triangle, $\text{Apex}(u|x)$, is parameterized (Liu 2008; Mendel et al. 2009) as ($w \in [0, 1]$):

$$\text{Apex}(u|x) = \underline{\mu}_{\tilde{A}}(x) + w[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)] \tag{6.53}$$

When[33] $w = 0$ the secondary MF is a right triangle whose right angle is perpendicular to $\underline{\mu}_{\tilde{A}}(x)$; when $w = 1/2$ the secondary MF is an isosceles triangle; and, when $\overline{\mu}_{\tilde{A}}(x)\, w = 1$ the secondary MF is a right triangle whose right angle is

[32]This conclusion is different from the one that is stated in Mendel (2014) because when that article was written it was unknown to the author how to squish an FOU. Thanks to others, it is now known how to do this.

[33]Almaraashi et al. (2016) refer to w as an *apex factor* and even let it be a function of x. When the latter is done, then there will be one apex factor at each discretized value of the primary variable, and this is no longer a parsimonious representation.

perpendicular to $\overline{\mu}_{\tilde{A}}(x)$. w is treated as a design parameter, but it is the *same* for all of the vertical slices.

During computations, one will need to use α-cuts of the triangle secondary MF. They are easily found (Exercise 6.15), as $(w \in [0,1])$:

$$\begin{cases} \tilde{A}(x)_\alpha = [a_\alpha(x), b_\alpha(x)] \\ a_\alpha(x) = \underline{\mu}_{\tilde{A}}(x) + w[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)]\alpha \\ b_\alpha(x) = \overline{\mu}_{\tilde{A}}(x) - (1-w)[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)]\alpha \end{cases} \tag{6.54}$$

Example 6.22 **(symmetrical trapezoid secondary MFs)**: The vertices of the base of each trapezoid are located at $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$, and its top is defined by left and right endpoints (see Fig. 6.25), $EP_l(u|x)$ and $EP_r(u|x)$, that are parameterized (Liu 2008; Mendel et al. 2009) as $(w \in [0,1])$:

$$EP_l(u|x) = \underline{\mu}_{\tilde{A}}(x) + \frac{1}{2}w[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)] \tag{6.55}$$

$$EP_r(u|x) = \overline{\mu}_{\tilde{A}}(x) - \frac{1}{2}w[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)] \tag{6.56}$$

When $w = 0$, the trapezoid reduces to a square well, and *the GT2 FS reduces to an IT2 FS*; and, when $w = 1$, $EP_l(u|x) = EP_r(u|x) = [\underline{\mu}_{\tilde{A}}(x) + \overline{\mu}_{\tilde{A}}(x)]/2$, so that the trapezoid reduces to an isosceles triangle. w is treated as a design parameter, but it is the *same* for all of the vertical slices.

During computations, one will need to use α-cuts of the symmetrical trapezoid secondary MF. They are also easily found (Exercise 6.16), as $(w \in [0,1])$:

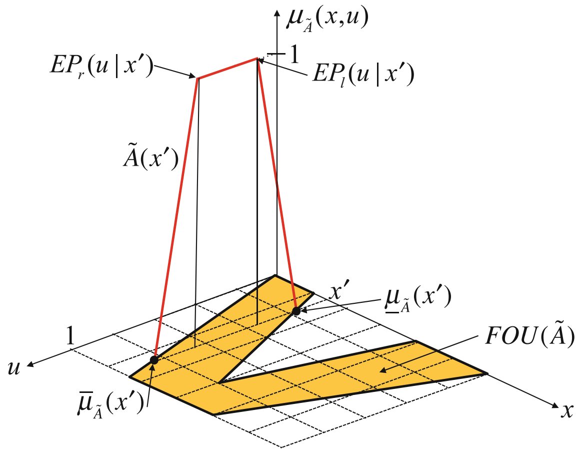


Fig. 6.25 Symmetrical trapezoid secondary MF for Example 6.22

$$\begin{cases} \tilde{A}(x)_\alpha = [a_\alpha(x), b_\alpha(x)] \\ a_\alpha(x) = \underline{\mu}_{\tilde{A}}(x) + \frac{1}{2}w[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)]\alpha \\ b_\alpha(x) = \overline{\mu}_{\tilde{A}}(x) - \frac{1}{2}w[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)]\alpha \end{cases} \tag{6.57}$$

Example 6.23 **(nonsymmetrical trapezoid secondary MFs)**: Another choice for a secondary MF is a nonsymmetrical trapezoid; however, it requires two parameters to define it. The vertices of the base of each trapezoid are located at $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$, and its top is defined by left and right endpoints (see Fig. 6.25), but $EP_l(u|x)$ and $EP_r(u|x)$ are now parameterized as:

$$EP_l(u|x) = \underline{\mu}_{\tilde{A}}(x) + w_1[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)] \tag{6.58}$$

$$EP_r(u|x) = \overline{\mu}_{\tilde{A}}(x) - w_2[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)] \tag{6.59}$$

$$\begin{cases} w_2 \neq w_1 \\ w_1 + w_2 < 1 \end{cases} \tag{6.60}$$

Equation (6.60) guarantees non-symmetry and $EP_r(u|x) > EP_l(u|x)$ (Exercise 6.17). When $w_1 = w_2 = 0$, the nonsymmetrical trapezoid reduces to a square well, and *the GT2 FS reduces to an IT2 FS*; when $w_1 = 0$, $EP_l(u|x) = \underline{\mu}_{\tilde{A}}(x)$ so that the non-symmetrical trapezoid has a vertical left leg; and, when $w_2 = 0$, $EP_r(u|x) = \overline{\mu}_{\tilde{A}}(x)$ so that the nonsymmetrical trapezoid has a vertical right leg. w_1 and w_2 are treated as design parameters, but they are the *same* for all of the vertical slices.

Exercise 6.17 also asks the reader to obtain formulas for the α-cuts of the nonsymmetrical trapezoid MF.

Our conclusion is that the *vertical-slice representation of a GT2 FS is a very parsimonious representation for the optimal design of a GT2 fuzzy system.*

6.8 Representing Non T2 FSs as T2 FSs

It may happen that a calculation involves a mixture of T1, IT2 and GT2 FSs, e.g., the union or intersection of such a mixture may be needed. This is easily accomplished by elevating a T1 FS to an IT2 FS. Nothing has to be done to an IT2 FS because it already is a special case of a GT2 FS.

Elevating a T1 FS, A, to an IT2 FS, $\tilde{A}$, is accomplished by assigning a secondary grade of 1 to A at $x \in X$, i.e.,

$$A \rightarrow \tilde{A} = \left\{ \left((x, u), \mu_{\tilde{A}}(x, u) = 1 \right) | x \in X,\ u = \mu_A(x) \right\} \tag{6.61}$$

This can also be summarized by means of the following expressive equation:

$$A \to \tilde{A} = 1/A \tag{6.62}$$

This expressive notation has already been used in (6.17) and (6.18).

When one wants to elevate an IVFS to an IT2 FS this is done by means of (6.18), i.e.,

$$\text{IVFS} \to \tilde{A} = 1/\text{IVFS} \tag{6.63}$$

Note that $1/\text{IVFS} = \text{CIT2 FS}$.

Definition 6.26 An element $x = x'$ is said (Mizumoto and Tanaka 1976) to have a *zero membership in a T2 FS* if it has a secondary grade equal to 1 corresponding to the primary membership of 0, and if it has all other secondary grades equal to 0. Such a 0 secondary membership at x' is denoted 1/0.

An element $x = x'$ is said to have a membership of one in a T2 FS—a *type-2 fuzzy singleton*—if it has a secondary grade equal to 1 corresponding to the primary membership of 1, and, if it has all other secondary grades equal to 0. Such a secondary membership at x' is denoted 1/1.

6.9 Returning to Linguistic Labels for T2 FSs

As explained in Sect. 2.2.5, sometimes it is necessary to go from MF numerical values for a variable to a linguistic description of that variable. This section examines how to do this for T2 FSs.

Consider the situation depicted in Fig. 6.26 at $x = x'$. This value of x only generates a nonzero membership in the T2 FS $\tilde{F}_4$ = *moderately high*; hence, $x = x'$ can be described linguistically, without any ambiguity, as "moderately high." The situation at $x = x''$ is quite different, because this value of x generates a range of

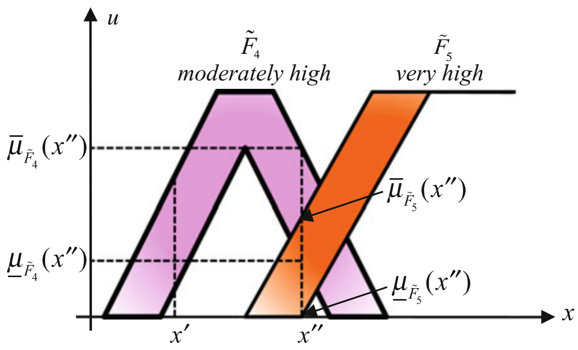


Fig. 6.26 Returning to a linguistic label for T2 FSs. Secondary MFs are not shown

nonzero secondary MF values in the two T2 FSs $\tilde{F}_4$ = *moderately high* and $\tilde{F}_5$ = *very high*. It would be extraordinarily difficult to communicate this linguistically. One approach, described next, is to first convert the intersection of the vertical line at $x = x''$ with the FOUs into a collection of numbers, after which the linguistic label at $x = x''$ can be chosen using an algorithm similar to the one for T1 FSs in (2.10).

The type-2 MFs $\mu_{\tilde{F}_4}(x,u)$ and $\mu_{\tilde{F}_5}(x,u)$ are described by $\mathrm{FOU}(\tilde{F}_4)$ and $\mathrm{FOU}(\tilde{F}_5)$, respectively, as well as by their secondary MFs. The upper and lower MFs for $\mathrm{FOU}(\tilde{F}_4)$ are $\overline{\mu}_{\tilde{F}_4}(x)$ and $\underline{\mu}_{\tilde{F}_4}(x)$, whereas the comparable quantities for $\mathrm{FOU}(\tilde{F}_5)$ are $\overline{\mu}_{\tilde{F}_5}(x)$ and $\underline{\mu}_{\tilde{F}_5}(x)$. Consider, for example, the vertical line at $x = x''$, and its intersections with $\mathrm{FOU}(\tilde{F}_4)$. Associated with the interval $[\underline{\mu}_{\tilde{F}_4}(x''), \overline{\mu}_{\tilde{F}_4}(x'')]$ is the secondary MF, $\mu_{\tilde{F}_4(x'')}(u)$, $u \in [\underline{\mu}_{\tilde{F}_4}(x''), \overline{\mu}_{\tilde{F}_4}(x'')]$. Let the center of gravity of the T1 FS $\mu_{\tilde{F}_4(x'')}(u)$ be denoted $u^{\mathrm{cg}}_{\tilde{F}_4(x'')}$. In a similar manner, $u^{\mathrm{cg}}_{\tilde{F}_5(x'')}$ can be computed so that $u^{\mathrm{cg}}_{\tilde{F}_4(x'')}$ and $u^{\mathrm{cg}}_{\tilde{F}_5(x'')}$ can then be compared. If $u^{\mathrm{cg}}_{\tilde{F}_4(x'')} > u^{\mathrm{cg}}_{\tilde{F}_5(x'')}$, then one would speak of x'' as "being moderately high"; otherwise one would speak of x'' as "being very high."

What has just been explained can be formalized as follows. Given p T2 FSs $\tilde{F}_i$ with MFs $\mu_{\tilde{F}_i}(x,u)$ that are characterized by $\mathrm{FOU}(\tilde{F}_i)$ $(i = 1,\ldots,p)$, whose upper and lower MFs, are $\overline{\mu}_{\tilde{F}_i}(x)$ and $\underline{\mu}_{\tilde{F}_i}(x)$, respectively, and by their secondary MFs. Consider an arbitrary value of x, say $x = x'$, and compute $\max[u^{\mathrm{cg}}_{\tilde{F}_1(x')}, u^{\mathrm{cg}}_{\tilde{F}_2(x')}, \ldots, u^{\mathrm{cg}}_{\tilde{F}_p(x')}] \equiv u^{\mathrm{cg}}_{\tilde{F}_m(x')}$, where $u^{\mathrm{cg}}_{\tilde{F}_i(x')}$ is the center of gravity of the secondary MF $\mu_{\tilde{F}_i(x')}(u)$, $u \in [\underline{\mu}_{\tilde{F}_i}(x'), \overline{\mu}_{\tilde{F}_i}(x')]$. Let $L(x')$ denote the linguistic label associated with x'. Then, $L(x') \equiv \tilde{F}_m$, i.e.,

$$L(x') = \arg\max_{\forall \tilde{F}_i} [u^{\mathrm{cg}}_{\tilde{F}_1(x')}, u^{\mathrm{cg}}_{\tilde{F}_2(x')}, \ldots, u^{\mathrm{cg}}_{\tilde{F}_p(x')}] \tag{6.64}$$

Example 6.24 For interval secondary MFs, it is easy to compute $u^{\mathrm{cg}}_{\tilde{F}_i(x')}$, as

$$u^{\mathrm{cg}}_{\tilde{F}_i(x')} = \frac{1}{2}\left[\overline{\mu}_{\tilde{F}_i}(x') + \underline{\mu}_{\tilde{F}_i}(x')\right] \tag{6.65}$$

For non-interval secondary MFs, the computations of the $u^{cg}_{\tilde{F}_i(x')}$ will be more complicated, and will, most likely, have to be done numerically.

Note that when T2 FSs reduce to T1 FSs, i.e., when $(x' \in X)$ $\overline{\mu}_{\tilde{F}_i}(x') = \underline{\mu}_{\tilde{F}_i}(x') = \mu_{F_i}(x')$, (6.64) reduces to (2.10). This is consistent with our basic design requirement that, when all MF uncertainties disappear, type-2 results must reduce to their well-established type-1 results.

Sometimes it is necessary to go from a complete FOU (for an IT2 FS) to a linguistic description of that variable. The approach that has just been described cannot be used to do this because it focuses on mapping only a single numerical value for a variable into a linguistic description of that variable. Similarity of IT2 FSs can be used to accomplish the former [e.g., see Mendel and Wu (2010, Chap. 4)].

6.10 Multivariable Membership Functions

All of the discussions in this chapter, so far, have been for T2 FSs that depend on only one variable. This section describes how to characterize T2 FSs that depend on up to p variables, $x_1, x_2, \ldots, x_p$.

To begin, our focus is on a T2 FS, $\tilde{A}$, that depends on only two variables, x_1 and x_2. When $X_1 \times X_2$ is continuous, then, analogous to (6.38) and (6.39), $\tilde{A}$ can be expressed as

$$\tilde{A} = \int_{x_1 \in X_1} \int_{x_2 \in X_2} \mu_{\tilde{A}(x_1,x_2)}(u)/(x_1, x_2) \tag{6.66}$$

where

$$\mu_{\tilde{A}(x_1,x_2)}(u) = \int_{u \in [0,1]} f_{(x_1,x_2)}(u)/u \tag{6.67}$$

When the multivariable secondary MF $\mu_{\tilde{A}(x_1,x_2)}(u)$, which is a T1 FS at each (x_1, x_2) pair, is[34] *separable* (see Sect. 2.14) then it can be expressed in terms of $\mu_{\tilde{A}(x_1)}(u)$ and $\mu_{\tilde{A}(x_2)}(u)$, as

$$\mu_{\tilde{A}(x_1,x_2)}(u) = \mu_{\tilde{A}(x_1)} \sqcap \mu_{\tilde{A}(x_2)} \tag{6.68}$$

where $\sqcap$ denotes the meet operation which is associated with computing the intersection of T2 FSs, and is discussed in great detail in Chap. 7. (6.68) is analogous to (2.115).

The extensions of these two-variable results to more than two variables is straightforward, e.g., for p variables, for which $\mu_{\tilde{A}(x_1,\ldots,x_p)}(u)$ is separable,

$$\mu_{\tilde{A}(x_1,\ldots,x_p)}(u) = \mu_{\tilde{A}(x_1)} \sqcap \mu_{\tilde{A}(x_2)} \sqcap \cdots \sqcap \mu_{\tilde{A}(x_p)} \tag{6.69}$$

Equation (6.69) is analogous to (2.116).

Note that (6.68) and (6.69) are widely used in Chaps. 7, 9 and 11, and how to actually compute them is explained in Chap. 7.

[34]To-date, there is no literature about non-separable multivariable secondary MFs.

Exercises

6.1 Provide an example that illustrates the difference between a DOU and an FOU.

6.2 As in Example 6.13, determine the FOU for a triangular MF (see Fig. 6.14), but for when $m_b - \sigma_b < m_a + \sigma_a$.

6.3 A possible alternative to asking a subject to provide interval end-point values for a linguistic term is to ask the subject to provide a *center location* (c) and *interval length* (l) for the interval.

(a) Formulate the question you would ask a subject so as to obtain these values.

(b) Determine the FOU when c and l information are obtained from a group of subjects. Assume that c has uncertainty $\pm\sigma_c$, l is centered about c, and that its uncertainty, $\pm\sigma_l$, is the same about its two endpoints $c - l/2$ and $c + l/2$. Consider two cases, when: (i) uncertainty interval for c, $m_c \pm \sigma_c$, does not overlap with the uncertainty intervals for the interval endpoints; and (ii) uncertainty interval for c, $m_c \pm \sigma_c$, overlaps with the uncertainty intervals for the interval endpoints.

6.4 Show three different FOUs that have lower and upper trapezoidal MFs.

6.5 Sketch $\mathrm{FOU}(\tilde{A})$ when, $\mathrm{UMF}(\tilde{A}) = \exp\left\{-\frac{1}{2}[(x-m)/\sigma]^2\right\}$, $\mathrm{LMF}(\tilde{A}) = s\exp\left\{-\frac{1}{2}[(x-m)/\sigma]^2\right\}$ and $0 < s < 1$.

6.6 Consider the combined case of Examples 6.16 and 6.17, i.e., a Gaussian primary MF with both an uncertain mean and standard deviation.

(a) Sketch its FOU.

(b) Determine formulas for its upper and lower MFs.

6.7 For Example 6.17, derive:

$$\partial N(x; m_1, \sigma)/\partial m_1 = (x - m_1)N(x; m_1, \sigma)/\sigma^2 \ (x < m_1)$$

$$\partial N(x; m_1, \sigma)/\partial \sigma = (x - m_1)^2 N(x; m_1, \sigma)/\sigma^3 \ (x < m_1)$$

6.8 Write formulas for:

(a) The LMF and the UMF of the FOU that is depicted in Fig. 6.9.

(b) The seven vertical slices that are also depicted on that figure.

6.9 Sketch four embedded T1 FSs for the type-2 MF that is depicted in:

(a) Fig. 6.16.

(b) Fig. 6.17.

6.10 Create an example that uses a mixture of a multiset from an IT2 FS and a multi interval-valued FS as an IT2 FS with the same level of detail as in Example 6.18 (figure and equations). Explain why such an IT2 FS can also be called a *multi interval-valued FS from an IT2 FS*.

6.11 Using the multiset from an IT2 FS that is depicted in Fig. 6.19:

   (a) At each x_i, sample u so that only the spikes are picked up. How many embedded T2 FSs will there be?
   (b) Enumerate all of the embedded T2 FSs.
   (c) Verify the truth of Theorem 6.2.
   (d) Verify the truth of Theorem 6.3.

6.12 Describe the minimal covering of $\mathrm{FOU}(\tilde{A})$ that is depicted in Fig. 6.27.

6.13 Describe the minimal covering of $\mathrm{FOU}(\tilde{B})$ in Fig. 6.28.

6.14 It has been cleverly suggested (Wu 2011) that an FOU can be covered by embedded T1 FSs that are both normal and convex, in which case there could be a "Constrained Representation Theorem" which would represent any IT2 FS as the union of such embedded T1 FSs. However, Wu (2011) provides a counterexample to this representation, so that such a theorem does not exist. In the FOU that is depicted in Fig. 6.29, shade in the region that cannot be covered by embedded T1 FSs that are both normal and convex.

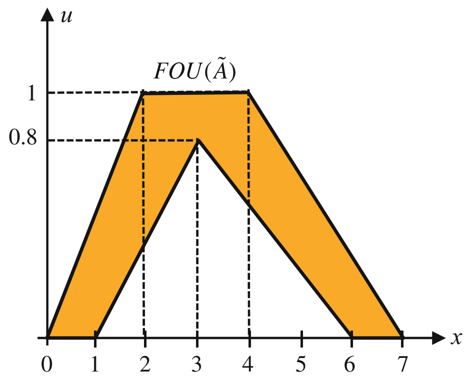


Fig. 6.27 $\mathrm{FOU}(\tilde{A})$ for Exercise 6.12 (Wu and Mendel 2007)

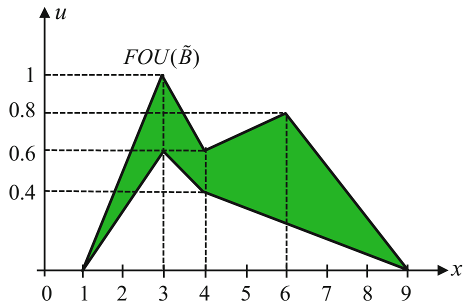
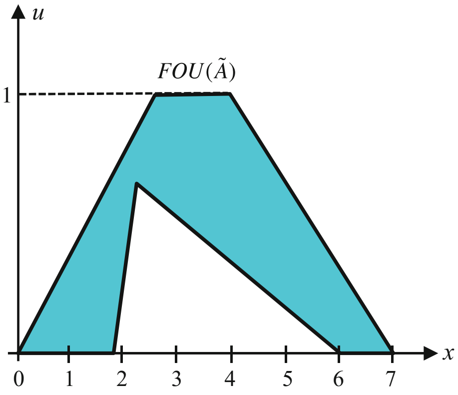


Fig. 6.28 $\mathrm{FOU}(\tilde{B})$ for Exercise 6.13

Fig. 6.29 FOU($\tilde{A}$) for Exercise 6.14



6.15 Derive the α-cut formulas that are in (6.54).

6.16 Derive the α-cut formulas that are in (6.57).

6.17 (a) Explain why the second line of (6.60) guarantees $EP_r(u|x) > EP_l(u|x)$. (b) Derive the α-cut formulas for the nonsymmetrical trapezoid in (6.58)–(6.60).

6.18 Explain how a squished FOU can be obtained by using two homotopies (see Example 6.19).

6.19 In this exercise (McCulloch and Wagner 2016), the lower and upper MFs of FOU($\tilde{A}$) are normal triangles, where a triangle MF for a T1 FS A is denoted $\mu_A(x) = \text{trimf}(x; [a, b, c; w])$ and

$$\mu_A(x) = \begin{cases} w(x-a)/(b-a) & a \le x \le b \\ w(c-x)/(c-b) & b < x \le c \\ 0 & \text{otherwise} \end{cases}$$

w is a weight that is used to adjust the height of the MF. In this exercise, $w = 1$, $\text{UMF}(\tilde{A}) = \text{trimf}(x; [\bar{a}, b, \bar{c}; 1])$, $\text{LMF}(\tilde{A}) = \text{trimf}(x; [\underline{a}, b, \underline{c}; 1])$, where $\bar{a} < \underline{a}$ and $\bar{c} > \underline{c}$, and $\alpha_k = k/k_{\max}$, $k = 0, 1, \ldots, k_{\max}$.

(a) Sketch FOU($\tilde{A}$).
(b) What are the formulas for $\bar{a}_k$, b_k, $\bar{c}_k$, $\underline{a}_k$ and $\underline{c}_k$ such that FOU($\tilde{A}_k$) is a squished version of FOU($\tilde{A}_{k-1}$), where the squishing is uniform as one goes from one level to the next?
(c) What is the squishing parameter?
(d) Is there a connection between the way in which FOU($\tilde{A}$) is squished in this exercise and the way in which it is squished in Exercise 6.18?

6.20 Another way to squish a horizontal slice at level α_1 into a horizontal-slice at level α_2 ($\alpha_2 > \alpha_1$) is to begin with the FOU (at level $\alpha_1 = 0$) and use the same UMF of the FOU at all α levels, but let the LMF of the horizontal slice at level $\alpha_2(\alpha_2 > \alpha_1)$ be a scaled version of the UMF of the FOU. Let γ_k denote the squishing parameter at level α_k.

(a) What is the formula for the $\alpha-$cut of the secondary MF for this GT2 FS?
(b) Sketch the secondary MF.

6.21 When $X \to X_d$ and $U \to U_d$, explain which definitions in this chapter can be easily modified by replacing X by X_d, U by U_d, and $\int$ by $\sum$.

6.22 When $X \to X_d$ but U remains a continuous universe of discourse, explain which definitions in this chapter can be easily modified, and how.

References

Aisbett, J., J.T. Rickard, and D.G. Morgenthaler. 2010. Type-2 fuzzy sets as functions on spaces. *IEEE Transactions on Fuzzy Systems* 18: 841–844.

Almaraashi, M., R. John, A. Hopgood, and S. Ahmadi. 2016. Learning of interval and general type-2 fuzzy logic systems using simulated annealing: Theory and practice. *Information Sciences* 360: 21–42.

Bilgin, A., H. Hagras, A. Malibari, M. Alhaddad, and D. Alghazzawi. 2012a. Towards a general type-2 fuzzy logic approach for computing with words using linear adjectives. In *Proceedings FUZZ-IEEE 2012*, pp. 1130–1137, Brisbane, AU.

Bilgin, A., J.Dooley, L. Whittington, H. Hagras, M. Henson, C. Wagner, A. Malibari, A. Al-Ghamdi, M. Alhaddad, and D. Alghazzawi. 2012b. Dynamic profile-selection for zslices based type-2 fuzzy agents controlling multi-user ambient intelligent environments. In *Proceedings FUZZ-IEEE 2012*, pp. 1392–1399, Brisbane, AU.

Bilgin, A., H. Hagras, A. Malibari, M. J. Alhaddad, and D. Alghazzawi. 2012c. A general type-2 fuzzy logic approach for adaptive modeling of perceptions for computing with words. In *Proceedings 2012 12th UK workshop on computational intelligence (UKCI)*, pp. 1–8.

Bilgin, A., H. Hagras, A. Malibari, M. J. Alhaddad, and D. Alghazzawi. 2013a. An experience based linear general type-2 fuzzy logic approach for computing with words. In *Proceedings of IEEE international conference on fuzzy systems*, Paper #1139, Hyderabad, India.

Bilgin, A., H. Hagras, A. Malibari, M. J. Alhaddad, and D. Alghazzawi. 2013b. Towards a linear general type-2 fuzzy logic based approach for computing with words. *International Journal of Soft Computing* 17:2203–2222.

Buckley, J.J. 2003. *Fuzzy probabilities: New approaches and new applications*. New York: Physica-Verlag.

Bustince, H. 2000. Indicator of inclusion grade for interval-valued fuzzy sets: Applications to approximate reasoning based on interval-valued fuzzy sets. *International Journal of Approximate Reasoning* 23: 137–209.

Bustince, H., J. Fernandez, H. Hagras, F. Herrera, M. Pagola, and E. Barrenechea. 2015. Interval type-2 fuzzy sets are generalization of interval-valued fuzzy sets: towards a wider view on their relationship. *IEEE Transactions on Fuzzy Systems* 23: 1876–1882.

Chen, Q., and S. Kawase. 2000. On fuzzy-valued fuzzy reasoning. *Fuzzy Sets and Systems* 113: 237–251.

Coupland, S., and R.I. John. 2007. Geometric type-1 and type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 15: 3–15.

Garibaldi, J. M., S. Musikasuwan, and T. Ozen. 2005. The association between non-stationary and interval type-2 fuzzy sets: A case study. In *Proceedings IEEE FUZZ conference*, pp. 224–229, Reno, NV.

Gorzalczany, M.B. 1987. A method of inference in approximate reasoning based on interval-valued fuzzy sets. *Fuzzy Sets and Systems* 21: 1–17.

Greenfield, S., and R. John. 2007. Optimized generalized type-2 join and meet operations. In *Proceedings FUZZ-IEEE 2007,* pp. 141–146, London, UK.

Hamrawi, H., and S. Coupland. 2009. Type-2 fuzzy arithmetic using alpha-planes. In *Proceedings IFSA/EUSFLAT*, pp. 606–611, Portugal.

Hamrawi, H., S. Coupland, and R. John. 2010. A novel alpha-cut representation for type-2 fuzzy sets. In *Proceedings FUZZ-IEEE 2010, IEEE world congress on computational intelligence*, pp. 351–358, Barcelona, Spain.

John, R., and S. Coupland. 2012. type-2 fuzzy logic: Challenges and misconceptions. *IEEE Computational Intelligence Magazine* 7 (3): 48–52.

Karnik, N.N., and J.M. Mendel. 1998. *An introduction to type-2 fuzzy logic systems*. USC-SIPI Report #418, Univ. of Southern Calif., Los Angeles, CA, June 1998. This can be accessed at: http://sipi.usc.edu/research; then choose “sipi technical reports/418.”.

Karnik, N.N., and J.M. Mendel. 2001a. Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems* 122: 327–348.

Karnik, N.N., and J.M. Mendel. 2001b. Centroid of a type-2 fuzzy set. *Information Sciences* 132: 195–220.

Karnik, N.N., J.M. Mendel, and Q. Liang. 1999. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 7: 643–658.

Klir, G.J., and B. Yuan. 1995. *Fuzzy sets and fuzzy logic: Theory and applications*. Upper Saddle River, NJ: Prentice Hall.

Kumbasar, T., and H. Hagras. 2015. A self-tuning zSlices based general type-2 fuzzy PI controller. *IEEE Transactions on Fuzzy Systems* 23: 991–1013.

Liang, Q., and J.M. Mendel. 2000. Interval type-2 fuzzy logic systems. In *Proceedings FUZZ-IEEE ’00*, San Antonio, TX.

Liu, F. 2008. An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. *Information Sciences* 178: 2224–2236.

Liu, F., and J.M. Mendel. 2008. Encoding words into interval type-2 fuzzy sets using an interval approach. *IEEE Transactions on Fuzzy Systems* 16: 1503–1521.

Ljung, L. 1999. *System identification: Theory for the user*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall.

Lushu, L. 1995. Random fuzzy sets and fuzzy martingales. *Fuzzy Sets and Systems* 69: 181–192.

McCulloch, J., and C. Wagner. 2016. Measuring the similarity between zSlices general type-2 fuzzy sets with non-normal secondary membership functions. In *Proceedings of FUZZ-IEEE 2016*, pp. 461–468, Vancouver, Canada.

Mendel, J.M. 2001. *Introduction to rule-based fuzzy logic systems*. Upper Saddle River, NJ: Prentice-Hall.

Mendel, J.M. 2007. Type-2 fuzzy sets and systems: An overview. *IEEE Computational Intelligence Magazine* 2: 20–29.

Mendel, J.M. 2009. On answering the question ‘Where do I start in order to solve a new problem involving interval type-2 fuzzy sets?’. *Information Sciences* 179: 3418–3431.

Mendel, J.M. 2010. Comments on ‘α-plane representation for type-2 fuzzy sets: Theory and applications’. *IEEE Transactions on Fuzzy Systems* 18: 229–230.

Mendel, J.M. 2012. Plotting 2-1/2 D figures for general type-2 fuzzy sets by hand or by powerpoint. In *Proceedings FUZZ-IEEE 2012*, pp. 1490-1497, Brisbane, AU.

Mendel, J.M. 2014. General type-2 fuzzy logic systems made simple: A tutorial. *IEEE Transactions on Fuzzy Systems* 22: 1162–1182.

Mendel, J.M., and R.I. John. 2002. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems* 10: 117–127.

Mendel, J.M., and Q. Liang. 1999. Pictorial comparisons of type-1 and type-2 fuzzy logic systems. In *Proceedings IASTED international conference on intelligent systems & control*, Santa Barbara, CA.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: Aiding people in making subjective judgments*. Hoboken, NJ: Wiley and IEEE Press.

Mendel, J.M., R.I. John, and F. Liu. 2006. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems* 14: 808–821.

Mendel, J.M., F. Liu, and D. Zhai. 2009. Alpha-plane representation for type-2 fuzzy sets: Theory and applications. *IEEE Transactions an Fuzzy Systems* 17: 1189–1207.

Mendel, J.M., M.R. Rajati, and P. Sussner. 2016. On clarifying some notations used for type-2 fuzzy sets as well as some recommended notational changes. *Information Sciences* 340–341: 337–345.

Mizumoto, M., and K. Tanaka. 1976. Some properties of fuzzy sets of type-2. *Information and Control* 31: 312–340.

Mizumoto, M., and K. Tanaka. 1981. Fuzzy sets of type-2 under algebraic product and algebraic sum. *Fuzzy Sets and Systems* 5: 277–290.

Moharrer, M., H. Tahayori, H., and A. Sadeghian. 2013. Modeling complex concepts with type-2 fuzzy sets: The case of user satisfaction of online services. In *Advances in type-2 fuzzy sets and systems: theory and applications*, ed. Sadeghian, et al., 133–146. NY: Springer.

Moller, B., and M. Beere. 2004. *Fuzzy randomness: Uncertainty in civil engineering and computational mechanics*. New York: Springer.

Pedrycz, W. 2015. Concepts and design aspects of granular models of type-1 and type-2. *International Journal of Fuzzy Logic and Intelligent Systems* 15: 87–95.

Rakshit, P., A. Chakraborty, A. Konar, and A.K. Nagar. 2013. Secondary membership evaluation in generalized type-2 fuzzy sets by evolutionary optimization algorithm. In *Proceedings FUZZ-IEEE 2013*, Paper #1334, Hyderabad, India.

Rakshit, P., A. Saha, A. Konar, and S. Saha. 2016. A type-2 fuzzy classifier for gesture recognition induced pathological disorder recognition. *Fuzzy Sets and Systems* 305: 95–130.

Starczewski, J.T. 2009a. Efficient triangular type-2 fuzzy logic systems. *International Journal of Approximate Reasoning* 50: 799–811.

Starczewski, J.T. 2009b. Extended triangular norms. *Information Sciences* 179: 742–757.

Tahayori, H., A.G.B. Tettamanzi, and G.D. Antoni. 2006. Approximated type-2 fuzzy set operations. In *Proceedings FUZZ-IEEE 2006*, Vancouver, B. C. Canada, pp. 9042–9049.

Tahayori, H., A.G.B. Tettamanzi, G.D. Antoni, A. Visconti, and M. Moharrer. 2010. Concave type-2 fuzzy sets: Properties and operations. *Soft Computing Journal* 14(7): 749–756.

Ulu, C., M. Güzellkaya, and I. Eksin. 2013. Granular type-2 membership functions: A new approach to formation of footprint of uncertainty in type-2 fuzzy sets. *Applied Soft Computing* 13: 3713–3728.

Wagner, C., and H. Hagras. 2008. z Slices–towards bridging the gap between interval and general type-2 fuzzy logic. In *Proceedings IEEE FUZZ conference*, Paper # FS0126, Hong Kong, China.

Wagner, C., and H. Hagras. 2010. Towards general type-2 fuzzy logic systems based on zSlices. *IEEE Transactions on Fuzzy Systems* 18: 637–660.

Wagner, C., and H. Hagras. 2013. zSlices based general type-2 fuzzy sets and systems. In *Advances in type-2 fuzzy sets and systems: Theory and applications*, eds. Sadeghian, A., J.M. Mendel, and H. Tahayori. New York: Springer.

Wu, D. 2011. A constrained representation theorem for interval type-2 fuzzy sets using convex and normal embedded type-1 fuzzy sets and its application to centroid computation. In *Proceedings of world conference on soft computing*, Paper #200, San Francisco, CA.

Wu, D., and J.M. Mendel. 2007. Uncertainty measures for interval type-2 fuzzy sets. *Information Sciences* 177: 5378–5393.

Zadeh, L.A. 1975. The concept of a linguistic variable and its application to approximate reasoning–1. *Information Sciences* 8: 199–249.

Chapter 7
Working with Type-2 Fuzzy Sets

7.1 Introduction and Guide for the Reader

Now that type-2 fuzzy sets (T2 FSs) as well as their different representations have been introduced one needs to learn how to work with them. By "to work" is meant "to perform set theoretic and other kinds of operations." It will be seen in this chapter that working with IT2 FSs is easy, working with GT2 FSs is more challenging, and there are different ways for working with GT2 FSs due to their vertical and horizontal slice representations. A lot of space in this chapter is devoted to computing the union, intersection, and complement of T2 FSs because they are the most widely used operations in a type-2 fuzzy system. The sections have been organized so that they (for the most part) parallel the organization of Chap. 2, beginning with Sect. 2.4.

In the first edition of this book, (Mendel 2001) greater emphasis was put on IT2 FSs then on GT2 FSs, because only IT2 fuzzy systems were covered in it. Much has happened during the past 16–17 years, including the emergence of the horizontal-slice representation for a GT2 FS which has prompted researchers to reexamine GT2 fuzzy systems. Consequently, some items about GT2 FSs that were in appendixes of Mendel (2001) are now in the main body of this chapter.

This chapter can be read in different ways depending upon the interests of the reader. Because the chapter is not easy reading, the following guide is provided for the reader:

- *For the reader who is only interested in IT2 FSs and subsequently IT2 fuzzy systems*:

    (1) Cover Theorems 7.1 (union of GT2 FSs), 7.4 (intersection of GT2 FSs), 7.8 (complement of a GT2 FS) and 7.7 (sifting); these theorems also introduce the definitions and symbols for the join, meet, and negation. Read Examples 7.1, 7.2, 7.7, and 7.12.

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_7

(2) Read Sects. 7.2.4.1, 7.2.4.3, 7.3, 7.6–7.9, 7.11.1, 7.13, 7.14, and the portions of Appendix 1 that are for IT2 FSs.

- *For the reader who is interested in both IT2 FSs and GT2 FSs and subsequently their comparable fuzzy systems, but is only interested in pursuing the horizontal-slice representation for GT2 FSs and GT2 fuzzy systems*:

(1) Cover Theorems 7.1 (union of GT2 FSs), 7.4 (intersection of GT2 FSs), 7.8 (complement of a GT2 FS) and 7.7 (sifting); these theorems also introduce the definitions and symbols for the join, meet, and negation. Read Examples 7.1, 7.2 and 7.7, and 7.12.

(2) Read Sects. 7.2.4, 7.3–7.9, 7.11–7.14, and Appendix 1.

- *For the reader who is interested in both IT2 FSs and GT2 FSs and their comparable fuzzy systems, and is interested in pursuing the vertical-slice and horizontal-slice representations for GT2 FSs and GT2 fuzzy systems*: Read the entire chapter.

7.2 Set-Theoretic Operations for GT2 FSs Computed Using the Extension Principle

Consider two GT2 FSs $\tilde{A}$ and $\tilde{B}$ that are expressed, using the (6.38) and (6.39) vertical-slice representations, as:

$$\tilde{A} = \int_{x\in X} \mu_{\tilde{A}(x)}(u)\Big/x = \int_{x\in X}\left[\int_{u\in[0,1]} f_x(u)/u\right]\Big/x \tag{7.1}$$

$$\tilde{B} = \int_{x\in X} \mu_{\tilde{B}(x)}(u)\Big/x = \int_{x\in X}\left[\int_{u\in[0,1]} g_x(u)/u\right]\Big/x \tag{7.2}$$

This section focuses on how to compute the union, intersection, and complement of such GT2 FSs using the Extension Principle. These computations are considerably more complicated than they are for T1 FSs because at each x the membership grade of each participating GT2 FS is a function instead of a single number.

Mizumoto and Tanaka (1976) were the first to study how to compute the union, intersection, and complement for GT2 FSs using the Extension Principle, and to establish the properties of the membership grades of GT2 FSs. They also (Mizumoto and Tanaka 1981) examined GT2 FSs under the operations of algebraic product and algebraic sum. Nieminen (1977) provided more detail about the algebraic structure of GT2 FSs. Karnik and Mendel (1998a, b), (2001) extended the works of Mizumoto and Tanaka and obtained new algorithms for performing union, intersection, and complement for GT2 FSs, also based on the Extension Principle.

7.2.1 Union of GT2 FSs

Theorem 7.1 (Mizumoto and Tanaka 1976) *The* union of the GT2 FSs $\tilde{A}$ *and* $\tilde{B}$ *is computed in vertical-slice format, as* ($x \in X$):

$$\tilde{A} \cup \tilde{B} \Leftrightarrow \mu_{(\tilde{A} \cup \tilde{B})_x}(u) \equiv \mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)} = \int_{v \in [0,1]} \int_{w \in [0,1]} f_x(v) \bigstar g_x(w) / (u = v \vee w) \tag{7.3}$$

where $\sqcup$ *denotes the so-called* join *operation*, $\bigstar$ *indicates minimum or product*, *and* $\vee$ *indicates maximum.*[1,2]

The use of the notation $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ in (7.3) to indicate the join between the type-1 secondary sets $\tilde{A}(x)$ and $\tilde{B}(x)$ is, of course, a *shorthand notation* for the operations in the right-most part of (7.3). It should be clear from (7.3) that, because $\mu_{(\tilde{A} \cap \tilde{B})_x}(u) \equiv \mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$, $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ is actually the MF of a T1 FS, namely $\mu_{\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}}(u)$.

Note that (7.3) is valid whether or not $\tilde{A}(x)$ and $\tilde{B}(x)$ are normal or convex; hence, it is a very general result that is not limited to closed GT2 FSs.

Proof See Appendix 2, Sect. 2.1.

---

[1]For discrete universe of discourse replace symbol $\int$ by the symbol $\sum$, and $[0, 1]$ by $\{0, \ldots, 1\}$. This also applies to (7.7) and (7.18).

[2]Mizumoto and Tanaka (1976), a highly recommended paper because of its clear details and carefully worked out examples, express (7.3) as: $\tilde{A} \cup \tilde{B} \Leftrightarrow \mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x) = \int_v \int_w f_x(v) \bigstar g_x(w)/(v \vee w)$. They use the same notations (A and B) for T1 and GT2 FSs, something that later changed as the type-2 field developed, which is why the now commonly used $\tilde{A}$ and $\tilde{B}$ have been used in the just-stated formula for $\tilde{A} \cup \tilde{B}$. Using our notation for a secondary MF (Definition 6.3), their $\mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x)$ has been reexpressed as $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$, and their $\mu_{\tilde{A} \cup \tilde{B}}(x)$ has been replaced by $\mu_{(\tilde{A} \cup \tilde{B})_x}(u)$.

Just as one uses $A \cup B$ to denote the union of the T1 FSs A and B, one should really use $\tilde{A}(x) \sqcup \tilde{B}(x)$, rather than $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$, to denote the join of the T1 FSs $\tilde{A}(x)$ and $\tilde{B}(x)$. This notational problem was already recognized in the last paragraph of Sect. 2 in Karnik and Mendel (2001). However, because a fuzzy set is equivalent to its MF, $\tilde{A}(x) \sqcup \tilde{B}(x) \Leftrightarrow \mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$, so $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ has been used in (7.3) rather than $\tilde{A}(x) \sqcup \tilde{B}(x)$. Using $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ also preserves the link between Mizumoto and Tanaka's notation, since $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ looks very similar to $\mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x)$.

For minimum t-norm and maximum disjunction, $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ in (7.3) can be expressed as (Bustince et al. 2016): $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)} = \sup\{\min(f_x(v), g_x(w)) | \max(v, w) = u\}$, $(v, w \in [0, 1])$.

What (7.3) says is that to perform the join between two secondary sets, $\tilde{A}(x)$ and $\tilde{B}(x)$, $u = v \vee w$ must be performed between every possible pair of primary membership variables v and w, such that $v \in [0,1]$ and $w \in [0,1]$, and that the secondary grade of $\mu_{(\tilde{A}\cup\tilde{B})_x}$ must be computed as the t-norm operation between the corresponding secondary grades of $\mu_{\tilde{A}(x)}$ and $\mu_{\tilde{B}(x)}$, $f_x(v)$ and $g_x(w)$, respectively. This must be done for $x \in X$ to obtain $\mu_{\tilde{A}\cup\tilde{B}}$.

If more than one combination of v and w gives the same point $v \vee w$, then in the join the one with the largest membership grade is kept. Suppose, for example, $v_1 \vee w_1 = \theta$ and $v_2 \vee w_2 = \theta$. Then, within the computations of (7.3), the terms $f_x(v_1)\bigstar g_x(w_1)/\theta + f_x(v_2)\bigstar g_x(w_2)/\theta$ would be obtained, where + denotes union. Combining these two terms at the *common* θ is a type-1 computation in which a t-conorm is used for the union. In this book, the maximum t-conorm is chosen, as suggested by Zadeh (1975) and Mizumoto and Tanaka (1976).

Example 7.1 (Karnik and Mendel 1998b) To illustrate the details of the calculations in (7.3), consider two T2 FSs $\tilde{A}$ and $\tilde{B}$, and a particular element x for which $U = \{0, 0.1, \ldots, 0.9, 1\}$. The secondary MFs in these two sets are $\mu_{\tilde{A}(x)}(u) = 0.5/0 + 0.7/0.1$ and $\mu_{\tilde{B}(x)}(u) = 0.3/0.4 + 0.9/0.8$. From (7.3), using the minimum t-norm, it follows that:

$$\begin{aligned}
\mu_{(\tilde{A}\cup\tilde{B})_x}(u) &= \mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)} = (0.5/0 + 0.7/0.1) \sqcup (0.3/0.4 + 0.9/0.8) \\
&= \frac{\min(0.5, 0.3)}{0 \vee 0.4} + \frac{\min(0.5, 0.9)}{0 \vee 0.8} + \frac{\min(0.7, 0.3)}{0.1 \vee 0.4} + \frac{\min(0.7, 0.9)}{0.1 \vee 0.8} \\
&= 0.3/0.4 + 0.5/0.8 + 0.3/0.4 + 0.7/0.8 \\
&= \max\{0.3, 0.3\}/0.4 + \max\{0.5, 0.7\}/0.8 \\
&= 0.3/0.4 + 0.7/0.8
\end{aligned} \tag{7.4}$$

Example 7.2 It can easily be shown that the join operation in (7.3) reduces to the original union operation in (2.14) when the GT2 FSs reduce to T1 FSs. For example, consider the join operation in (7.3) when minimum t-norm is used. In the case of T1 FSs, $f_x(v)$ must have a value equal to 1 [see (6.62)] at only one value of v, say v_1, and $g_x(w)$ must also have a value equal to 1 at only one value of w, say w_1. All of the other $f_x(v)$ $(v \neq v_1)$ and $g_x(w)$ $(w \neq w_1)$ will be zero. Consequently, when the minima between all $f_x(v)$ and $g_x(w)$ are taken $(v \in [0,1]$ and $w \in [0,1])$, the only pair that gives a non-zero value is $\{f_x(v_1), g_x(w_1)\}$, and its minimum value is 1. So the union of the two sets will consist of only one element $v_1 \vee w_1 = \max\{v_1, w_1\}$, which agrees with (2.14).

Theorem 7.2 (Mizumoto and Tanaka 1976) *If* $\tilde{A}(x)$ *and* $\tilde{B}(x)$ *are normal convex T1 FSs, then* $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ *is also normal and convex.*

Proof While the proof of the normality of $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ is straightforward, and follows from the normality of $\tilde{A}(x)$ and $\tilde{B}(x)$, the proof of convexity of $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ is much more challenging. It is found in Mizumoto and Tanaka (1976, pp. 326–328).

Next, our attention is directed at how to compute $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ for continuous universes of discourse. Readers who are only interested in working with IT2 FSs can go directly to Sect. 7.2.2, because Theorem 7.3 will not be needed for IT2 FSs.

In the following, the notation is greatly simplified because the join involves T1 FSs, i.e., $F_1 \equiv \tilde{A}(x)$, $F_2 \equiv \tilde{B}(x)$, $\mu_{\tilde{A}(x)} \equiv f_1(\theta)$ and $\mu_{\tilde{B}(x)} \equiv f_2(\theta)$ (θ plays the role of the secondary variable u). Additionally, instead of using the notation $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$, the notation $F_1 \sqcup F_2$ is used, since it connects directly with the notation that is used in Karnik and Mendel (2001) (see also Footnote 2).

Theorem 7.3 (Karnik and Mendel 2001) (a) *Let F_1 and F_2 be two convex, normal T1 FSs, with MFs $f_1(\theta)$ and $f_2(\theta)$, respectively, and ξ_1 and ξ_2 be real numbers such that $\xi_1 \le \xi_2$ and $f_1(\xi_1) = f_2(\xi_2) = 1$. The MF of the join of F_1 and F_2 is:*

$$\mu_{F_1 \sqcup F_2}(\theta) = \begin{cases} f_1(\theta) \bigstar f_2(\theta) & \theta < \xi_1 \\ f_2(\theta) & \xi_1 \le \theta < \xi_2 \\ f_1(\theta) \vee f_2(\theta) & \theta \ge \xi_2 \end{cases} \tag{7.5}$$

where $\bigstar$ denotes the t-norm operation used, either minimum or product, and $\vee$ denotes maximum.

(b) *Let $F_1, \ldots,$ and F_n be* n *convex, normal, T1 FSs characterized by MFs $f_1(\theta), \ldots, f_n(\theta)$, respectively, and $\xi_1, \ldots, \xi_n$ be real numbers such that $\xi_1 \le \xi_2 \le \cdots \le \xi_n$ and $f_1(\xi_1) = \cdots = f_n(\xi_n) = 1$. Then the MF of $\sqcup_{i=1}^n F_i$ is:*

$$\mu_{\sqcup_{i=1}^n F_i}(\theta) = \begin{cases} T_{i=1}^n f_i(\theta) & \theta < \xi_1 \\ T_{i=k+1}^n f_i(\theta) & \xi_k \le \theta < \xi_{k+1} \quad 1 \le k \le n-1 \\ \vee_{i=1}^n f_i(\theta) & \theta \ge \xi_n \end{cases} \tag{7.6}$$

where T denotes the t-norm operation used, either minimum or product.

Note that: (a) the conditions "$\xi_1 \le \xi_2$ and $f_1(\xi_1) = f_2(\xi_2) = 1$" mean that the maximum value of $f_1(\theta)$ occurs either to the left of or at the maximum value of $f_2(\theta)$; and (b) Dubois and Prade (1978) present the same result given in part (a) of this theorem, but just for the minimum t-norm.

Proof See Appendix 2, Sect. 2.2.

Example 7.3 Figure 7.1b depicts the result of the join operation on the four Gaussian secondary MFs that are depicted in Fig. 7.1a, as obtained using (7.6) when $n = 4$ and the product t-norm. Although not marked on this figure, it is true that: $\xi_1 = 0.1$ [for $f_1(\theta)$], $\xi_2 = 0.25$ [for $f_2(\theta)$], $\xi_3 = 0.35$ [for $f_3(\theta)$] and $\xi_4 = 0.435$ [for $f_4(\theta)$]. So, when:

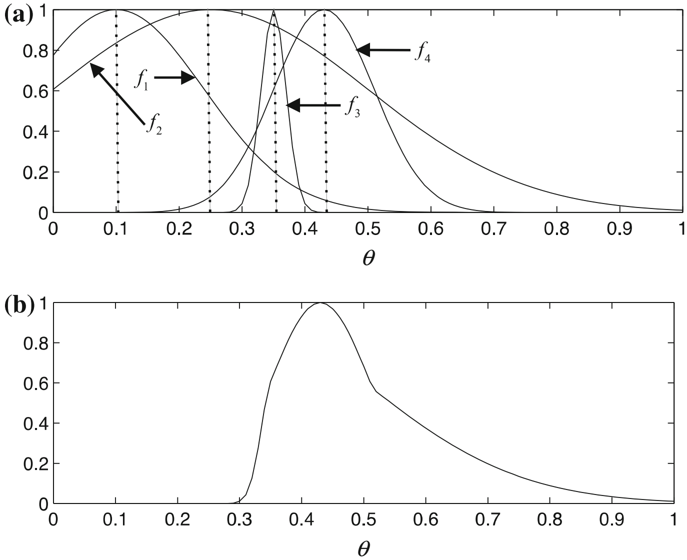


Fig. 7.1 Illustration of join operation for **a** four Gaussian secondary MFs, and **b** join under product t-norm

- $\theta < 0.1$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = f_1(\theta) \times f_2(\theta) \times f_3(\theta) \times f_4(\theta) \approx 0$
- $0.1 \le \theta < 0.25$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = f_2(\theta) \times f_3(\theta) \times f_4(\theta) \approx 0$
- $0.25 \le \theta < 0.35$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = f_3(\theta) \times f_4(\theta)$
- $0.35 \le \theta < 0.435$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = f_4(\theta)$
- $\theta \ge 0.435$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = \max_{i=1,\ldots,4} f_i(\theta) = \max[f_2(\theta), f_4(\theta)]$

Example 7.4 Figure 7.2b depicts the result of the join operation on the four Gaussian secondary MFs depicted in Fig. 7.2a, obtained using (7.6) when $n = 4$ and the minimum t-norm. Although not marked on this figure, it is true that: $\xi_1 = 0.1$ [for $f_1(\theta)$], $\xi_2 = 0.25$ [for $f_2(\theta)$], $\xi_3 = 0.4$ [for $f_3(\theta)$] and $\xi_4 = 0.5$ [for $f_4(\theta)$]. So, when:

- $\theta < 0.1$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = \min[f_1(\theta), f_2(\theta), f_3(\theta), f_4(\theta)] = f_4(\theta) \approx 0$
- $0.1 \le \theta < 0.25$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = \min[f_2(\theta), f_3(\theta), f_4(\theta)] = f_4(\theta) \approx 0$
- $0.25 \le \theta < 0.4$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = \min[f_3(\theta), f_4(\theta)] = f_4(\theta)$
- $0.4 \le \theta < 0.5$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = f_4(\theta)$
- $\theta \ge 0.5$, $\mu_{\sqcup_{i=1}^4 F_i}(\theta) = \max_{i=1,\ldots,4} f_i(\theta) = \max[f_3(\theta), f_4(\theta)]$

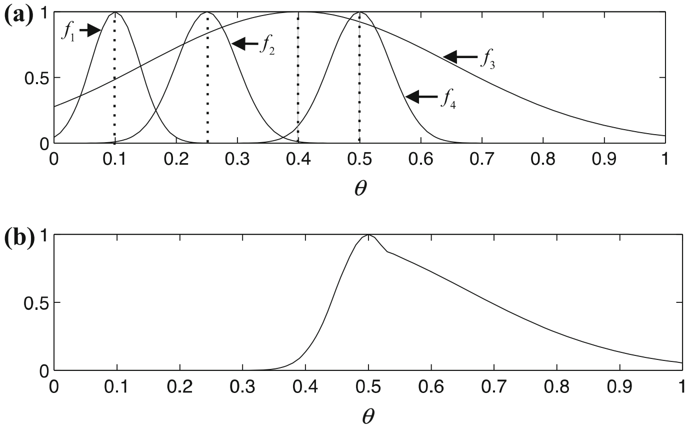


Fig. 7.2 Illustration of join operation for **a** four Gaussian secondary MFs, and **b** join under minimum t-norm

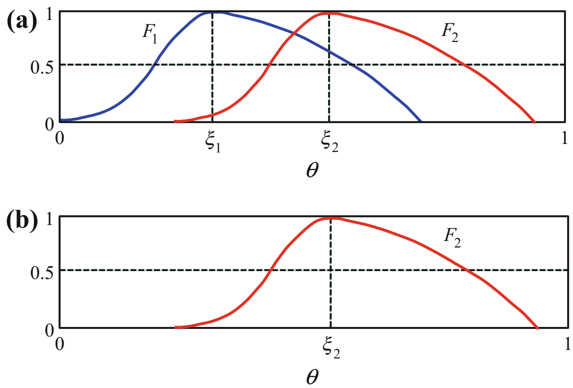


Fig. 7.3 Illustration of Corollary 7.1. **a** Convex and normal T1 FSs, whose MFs are shifted versions of each other, and **b** $F_1 \sqcup F_2 = F_2$

Corollary 7.1 (Karnik and Mendel 2001) *Let* $F_1, \ldots,$ *and* F_n *be* n *convex*, *normal*, *T1 FSs characterized by MFs* $f_1(\theta), \ldots,$ *and* $f_n(\theta)$, *respectively*, *such that* $f_i(\theta) = f_1(\theta - k_i)$ *and* $0 = k_1 \le k_2 \le \cdots \le k_n$, *then, under the* minimum t-norm, $\sqcup_{i=1}^{n} F_i = F_n$.

Proof The proof follows by a direct application of Theorem 7.3 to the n T1 FSs described in the statement of this corollary.

Example 7.5 Figure 7.3a depicts MFs for T1 FSs F_1 and F_2, where the MF of F_2 is a right-shifted duplicate MF of F_1. Using Corollary 7.1, it follows that, under minimum t-norm, $F_1 \sqcup F_2 = F_2$, as is depicted in Fig. 7.3b.

Example 7.6 Figure 7.4b is an example of the union under minimum t-norm of the two Gaussian GT2 FSs depicted in Fig. 7.4a. In Fig. 7.4a, if one draws a vertical

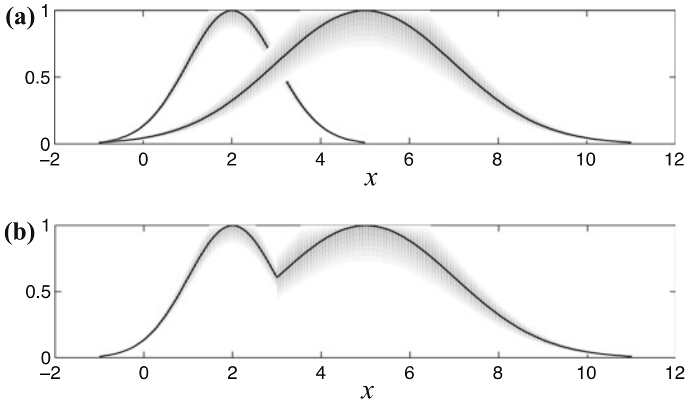


Fig. 7.4 Union of Gaussian T2 FSs. **a** Participating sets (heavy curves are the principal MFs), and **b** their union [heavy curve is the union of the principal MFs in (**a**)] (Karnik and Mendel 2001, © *Fuzzy Sets and Systems*, 2001)

line at any x-value along the horizontal axis the secondary MFs of the two participating Gaussian GT2 FSs are obtained. These secondary MFs are themselves Gaussian T1 FSs confined to the interval [0, 1]. Theorem 7.3a was applied to these T1 FSs to obtain the results for the join depicted in Fig. 7.4b.

Observe that the principal MF (Definition 6.13) of the result of the union operation can be obtained by performing that operation on the principal MFs of the participating GT2 FSs. Consequently, if one replaces all the GT2 FSs by T1 FSs, whose type-1 MFs are the principal MFs of the GT2 FSs, the type-2 union operation collapses to the correct type-1 operation.

Comparing Theorems 7.1 and 7.3, it is important to observe that: (1) they are both valid for $\bigstar$ = minimum or product and $\vee$ = maximum; (2) Theorem 7.1 is valid for $\tilde{A}(x)$ and $\tilde{B}(x)$ that do not have to be normal and convex; but, (3) Theorem 7.3 is valid only for $\tilde{A}(x)$ and $\tilde{B}(x)$ that are normal and convex. Consequently, Theorem 7.1 is more general than Theorem 7.3.

Finally, a totally different way to compute the union of GT2 FSs is described in Sect. 7.4.1. It uses the horizontal-slice representation of a GT2 FS, and, in the opinion of this author, is a simpler way to compute the union of GT2 FSs then what has been covered in this section.

7.2.2 *Intersection of GT2 FSs*

Theorem 7.4 (Mizumoto and Tanaka 1976) *The* intersection of the GT2 FSs $\tilde{A}$ *and* $\tilde{B}$ *is computed in vertical-slice format, as* $(x \in X)$:

$$\tilde{A} \cap \tilde{B} \Leftrightarrow \mu_{(\tilde{A} \cap \tilde{B})_x}(u) \equiv \mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)} = \int_{v \in [0,1]} \int_{w \in [0,1]} f_x(v) \bigstar g_x(w)/(u = v \wedge w) \tag{7.7}$$

where $\sqcap$ *denotes the so-called* meet *operation,*[3] $\bigstar$ *indicates minimum or product, and* $\wedge$ *indicates minimum.*

The use of the notation $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ to indicate the meet between the type-1 secondary sets $\tilde{A}(x)$ and $\tilde{B}(x)$ is, of course, another *shorthand notation*, but this time for the operations in the right-most part of (7.7). Footnote 1 also applies to (7.7). It should be clear from (7.7) that, because $\mu_{(\tilde{A} \cap \tilde{B})_x}(u) \equiv \mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$, $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ is actually the MF of a T1 FS, namely $\mu_{\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}}(u)$.

Note that (7.7) is also valid whether or not $\tilde{A}(x)$ and $\tilde{B}(x)$ are normal or convex; hence, like (7.3), it is a very general result. Note, also, that although $\wedge$ can in general be the minimum or product, in (7.7) it is only the minimum. The situation when $\wedge$ is the product is considered below beginning with (7.11).

Proof See Appendix 2, Sect. 2.3.

What (7.7) says is that to perform the meet between two secondary MFs, $\tilde{A}(x)$ and $\tilde{B}(x)$, $u = v \wedge w$ must be performed between every possible pair of primary membership variables v and w, such that $v \in [0, 1]$ and $w \in [0, 1]$, and that the secondary grade of $\mu_{(\tilde{A} \cap \tilde{B})_x}$ must be computed as the t-norm operation between the corresponding secondary grades of $\mu_{\tilde{A}(x)}$ and $\mu_{\tilde{B}(x)}$, $f_x(v)$ and $g_x(w)$, respectively. This must be done for $x \in X$ to obtain $\mu_{\tilde{A} \cap \tilde{B}}$.

If more than one combination of v and w gives the same point $v \wedge w$, then in the meet (just as in the join) the one with the largest membership grade is kept.

Example 7.7 This is a continuation of Example 7.1, in which $U = \{0, 0.1, \ldots, 0.9, 1\}$, $\mu_{\tilde{A}(x)}(u) = 0.5/0 + 0.7/0.1$ and $\mu_{\tilde{B}(x)}(u) = 0.3/0.4 + 0.9/0.8$. From (7.7), using the minimum t-norm, it follows that:

---

[3]Mizumoto and Tanaka (1976) express (7.7) as: $\tilde{A} \cap \tilde{B} \Leftrightarrow \mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x) = \int_v \int_w f_x(v) \bigstar g_x(w)/(v \bigstar w)$. Using our notation for a secondary MF (Definition 6.3), their $\mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x)$ is reexpressed as $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$.

Just as one uses $A \cap B$ to denote the intersection of the T1 FSs A and B, one should really use $\tilde{A}(x) \sqcap \tilde{B}(x)$, rather than $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$, to denote the intersection of the T1 FSs $\tilde{A}(x)$ and $\tilde{B}(x)$. However, because a fuzzy set is equivalent to its MF, $\tilde{A}(x) \sqcap \tilde{B}(x) \Leftrightarrow \mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$, so $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ has been used in (7.7) rather than $\tilde{A}(x) \sqcap \tilde{B}(x)$. Using $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ again preserves the link between Mizumoto and Tanaka's notation, since $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ looks very similar to $\mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x)$. For minimum t-norm and minimum conjunction, $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ in (7.7) can be expressed as Bustince et al. (2016): $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)} = \sup\{\min(f_x(v), g_x(w)) | \min(v, w) = u\}$, $(v, w \in [0, 1])$.

$$\begin{aligned}\mu_{(\tilde{A}\cap\tilde{B})_x}(u) &= \mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)} = (0.5/0 + 0.7/0.1) \sqcap (0.3/0.4 + 0.9/0.8) \\ &= \frac{\min(0.5, 0.3)}{0 \wedge 0.4} + \frac{\min(0.5, 0.9)}{0 \wedge 0.8} + \frac{\min(0.7, 0.3)}{0.1 \wedge 0.4} + \frac{\min(0.7, 0.9)}{0.1 \wedge 0.8} \\ &= 0.3/0 + 0.5/0 + 0.3/0.1 + 0.7/0.1 \\ &= \max\{0.3, 0.5\}/0 + \max\{0.3, 0.7\}/0.1 \\ &= 0.5/0 + 0.7/0.1\end{aligned} \tag{7.8}$$

Theorem 7.5 (Mizumoto and Tanaka 1976) *If $\tilde{A}(x)$ and $\tilde{B}(x)$ are normal convex T1 FSs, then $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ is also normal and convex.*

Proof Similar to the proof of Theorem 7.2 (Mizumoto and Tanaka 1976, pp. 326–328).

Next, our attention is directed at how to compute $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ for continuous universes of discourse. Readers who are only interested in working with IT2 FSs can go directly to Sect. 7.2.3, because Theorem 7.6 will not be needed for IT2 FSs. As in Sect. 7.2.1, $F_1 \equiv \tilde{A}(x)$, $F_2 \equiv \tilde{B}(x)$, $\mu_{\tilde{A}(x)} \equiv f_1(\theta)$, $\mu_{\tilde{B}(x)} \equiv f_2(\theta)$, and θ plays the role of the secondary variable u.

Theorem 7.6 (Karnik and Mendel 2001) (a) *Let F_1 and F_2 be two convex, normal T1 FSs, with MFs $f_1(\theta)$ and $f_2(\theta)$, respectively, and ξ_1 and ξ_2 be real numbers such that $\xi_1 \le \xi_2$ and $f_1(\xi_1) = f_2(\xi_2) = 1$. The MFs of the meet of F_1 and F_2 using the minimum t-norm is*:

$$\mu_{F_1 \sqcap F_2}(\theta) = \begin{cases} f_1(\theta) \vee f_2(\theta) & \theta < \xi_1 \\ f_1(\theta) & \xi_1 \le \theta < \xi_2 \\ f_1(\theta) \wedge f_2(\theta) & \theta \ge \xi_2 \end{cases} \tag{7.9}$$

where $\vee$ denotes maximum and $\wedge$ denotes the minimum.

(b) *Let $F_1, \ldots,$ and F_n be* n *convex, normal, T1 FSs characterized by MFs $f_1(\theta), \ldots,$ and $f_n(\theta)$, respectively, and $\xi_1, \ldots, \xi_n$ be real numbers such that $\xi_1 \le \xi_2 \le \cdots \le \xi_n$ and $f_1(\xi_1) = \cdots = f_n(\xi_n) = 1$. Then the MF of $\sqcap_{i=1}^n F_i$ is*:

$$\mu_{\sqcap_{i=1}^n F_i}(\theta) = \begin{cases} \vee_{i=1}^n f_i(\theta) & \theta < \xi_1 \\ \wedge_{i=1}^k f_i(\theta) & \xi_k \le \theta < \xi_{k+1} \quad 1 \le k \le n-1 \\ \wedge_{i=1}^n f_i(\theta) & \theta \ge \xi_n \end{cases} \tag{7.10}$$

Proof See Appendix 2, Sect. 2.4.

Example 7.8 This is a continuation of Example 7.3. Figure 7.5 depicts the result of the meet operation on the four Gaussian secondary MFs that are depicted in

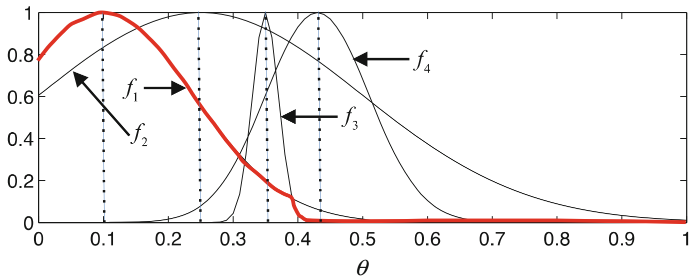


Fig. 7.5 Illustration of meet operation (*in red*) for the Examples 7.3 and 7.8 four Gaussian secondary MFs

Fig. 7.1a, as obtained using (7.10) when $n = 4$. Recall that: $\xi_1 = 0.1$ [for $f_1(\theta)$], $\xi_2 = 0.25$ [for $f_2(\theta)$], $\xi_3 = 0.35$ [for $f_3(\theta)$], and $\xi_4 = 0.435$ [for $f_4(\theta)$]. So, when:

- $\theta < 0.1,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \max[f_1(\theta), f_2(\theta), f_3(\theta), f_4(\theta)] = f_1(\theta)$
- $0.1 \le \theta < 0.25,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \min[f_1(\theta)] = f_1(\theta)$
- $0.25 \le \theta < 0.35,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \min[f_1(\theta), f_2(\theta)] = f_1(\theta)$
- $0.35 \le \theta < 0.435,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \min[f_1(\theta), f_2(\theta), f_3(\theta)] = \min[f_1(\theta), f_3(\theta)]$
- $\theta \ge 0.435,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \min_{i=1,\ldots,4} f_i(\theta) = f_3(\theta) \approx 0$

Example 7.9 This is a continuation of Example 7.4. Figure 7.6 depicts the meet operation on the four Gaussian secondary MFs that are depicted in Fig. 7.2a, obtained using (7.10) when $n = 4$. Recall that: $\xi_1 = 0.1$ [for $f_1(\theta)$], $\xi_2 = 0.25$ [for $f_2(\theta)$], $\xi_3 = 0.4$ [for $f_3(\theta)$], and $\xi_4 = 0.5$ [for $f_4(\theta)$]. So, when:

- $\theta < 0.1,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \max[f_1(\theta), f_2(\theta), f_3(\theta), f_4(\theta)] = \max[f_1(\theta), f_3(\theta)]$
- $0.1 \le \theta < 0.25,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \min[f_1(\theta)] = f_1(\theta)$
- $0.25 \le \theta < 0.4,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \min[f_1(\theta), f_2(\theta)] = f_1(\theta) \approx 0$
- $0.4 \le \theta < 0.5,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \min[f_1(\theta), f_2(\theta), f_3(\theta)] = f_1(\theta) \approx 0$
- $\theta \ge 0.5,\ \mu_{\sqcap_{i=1}^4 F_i}(\theta) = \min_{i=1,\ldots,4} f_i(\theta) = f_1(\theta) \approx 0$

Corollary 7.2 (Karnik and Mendel 2001) *Let* $F_1, \ldots,$ and F_n *be n convex, normal, T1 FSs characterized by MFs* $f_1(\theta), \ldots,$ and $f_n(\theta)$, *respectively, such that* $f_i(\theta) = f_1(\theta - k_i)$ *and* $0 = k_1 \le k_2 \le \cdots \le k_n$, *then, under the* minimum t-norm, $\sqcap_{i=1}^n F_i = F_1$.

Proof The proof follows by a direct application of Theorem 7.6 to the n type-1 sets described in the statement of this corollary.

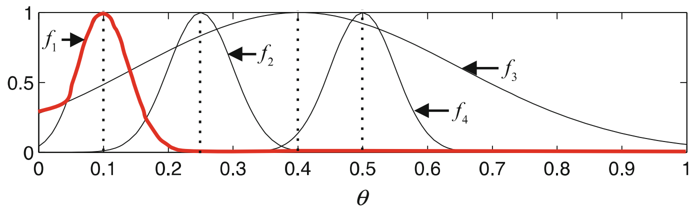


Fig. 7.6 Illustration of meet operation (*in red*) for the Examples 7.4 and 7.9 four Gaussian secondary MFs

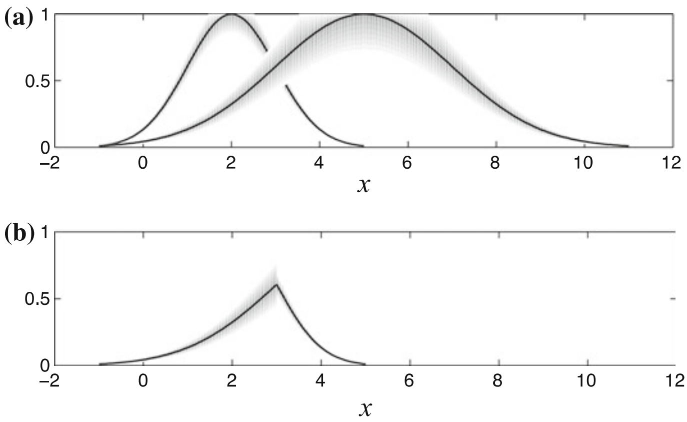


Fig. 7.7 Intersection of Gaussian GT2 FSs **a** Participating sets (heavy curves are the principal MFs), and **b** their intersection [heavy curve is the intersection of the principal MFs in (**a**)] (Karnik and Mendel 2001, © *Fuzzy Sets and Systems*, 2001)

Example 7.10 This is a continuation of Example 7.5. For F_1 and F_2 depicted in Fig. 7.3a, and from Corollary 7.2, it follows that $F_1 \sqcap F_2 = F_1$.

Example 7.11 This is a continuation of Example 7.6, and is an example of the intersection under minimum t-norm of the two Gaussian GT2 FSs depicted in Fig. 7.4a (repeated for the convenience of the readers in Fig. 7.7a. Theorem 7.6a was applied to the type-1 secondary sets to obtain the results for the meet depicted in Fig. 7.7b.

Observe that the principal MF of the result of the intersection operation can be obtained by performing that operation on the principal MFs of the participating GT2 FSs. Consequently, if one replaces all the GT2 FSs by T1 FSs whose type-1 MFs are the principal MFs of the GT2 FSs the type-2 intersection operation collapses to the correct type-1 operation.

Comparing Theorems 7.4 and 7.6, it is important to observe that: (1) they are both valid for $\wedge$ = minimum; (2) Theorem 7.4 is valid for $\bigstar$ = minimum or product; (3) Theorem 7.6 is valid only for $\bigstar$ = minimum; (4) Theorem 7.4 is valid for $\tilde{A}(x)$ and $\tilde{B}(x)$ that do not have to be normal and convex; but, (5) Theorem 7.6 is valid only for $\tilde{A}(x)$ and $\tilde{B}(x)$ that are normal and convex. Consequently, Theorem 7.4 is much more general than Theorem 7.6.

A totally different way to compute the intersection of GT2 FSs is described in Sect. 7.4.2. It uses the horizontal-slice representation of a GT2 FS, and, in the opinion of this author, is a simpler way to compute the intersection of GT2 FSs then what has been covered in this section.

Next, our attention is directed at the *meet operation using the product t-norm and conjunction*. As before, F_1 and F_2 are two convex, normal, T1 FSs that are characterized by MFs $f_1(\theta)$ and $f_2(\theta)$, respectively; hence, under the product t-norm and conjunction, (7.7) becomes:

$$F_1 \sqcap F_2 = \int_{v\in[0,1]} \int_{w\in[0,1]} [f_1(v)f_2(w)]/(vw) \tag{7.11}$$

This equation involves the product of secondary grades $f_1(v)$ and $f_2(w)$, as well as the product of v and w; hence, the analysis of the meet operation under product t-norm is quite different than that of the join or meet operations previously described.

If θ ($\theta \in \Re$) is an element of $F_1 \sqcap F_2$, then the membership grade of θ can be found by: (1) finding all the pairs $\{v, w\}$ such that $v \in [0, 1]$, $w \in [0, 1]$, and $vw = \theta \in [0, 1]$; (2) multiplying the secondary grades of v and w in each pair; and, (3) then finding the maximum of these products of secondary grades. The possible admissible $\{v, w\}$ pairs, whose product is θ, are $\{v, \theta/v\}$ $(v \in [0, 1],\ v \neq 0)$ for $\theta \neq 0$, and $\{v, 0\}$ or $\{0, w\}$ ($v \in [0, 1]$ and $w \in [0, 1]$) for $\theta = 0$. The products of the secondary grades of v and w are found from each such pair after which the maximum of all these products is taken as the membership grade of, i.e.

$$\mu_{F_1 \sqcap F_2}(\theta) = \sup_{v\in[0,1], v\neq 0} f_1(v)f_2(\theta/v) \quad \theta \in \Re \text{ and } \theta \neq 0 \tag{7.12}$$

and when $\theta = 0$

$$\mu_{F_1 \sqcap F_2}(0) = \left[\sup_{v\in[0,1]} f_1(v)f_2(0)\right] \vee \left[\sup_{w\in[0,1]} f_1(0)f_2(w)\right] \tag{7.13}$$

Observe that, because F_1 and F_2 are normal:

$$\sup_{v\in[0,1]} f_1(v)f_2(0) = f_2(0) \sup_{v\in[0,1]} f_1(v) = f_2(0) \times 1 = f_2(0) \tag{7.14}$$

and, similarly

$$\sup_{w \in [0,1]} f_1(0) f_2(w) = f_1(0) \tag{7.15}$$

In summary, for two convex, normal T1 FSs F_1 and F_2, *the meet under product t-norm can be expressed, when* $v \neq 0$ *as* (7.12), *and, when* $\theta = 0$ *as*

$$\mu_{F_1 \sqcap F_2}(0) = f_1(0) \vee f_2(0) \tag{7.16}$$

Note that, if $\theta/v = w$ is substituted into (7.12) a similar expression is obtained in terms of $f_1(\theta/w) f_2(w)$. Because the meet operation is commutative for the product t-norm (see Table 7.2), the same result is obtained whether one substitutes $\theta/w = v$ or $\theta/v = w$.

As is apparent from (7.12), the meet operation under product t-norm is very much dependent on the functions $f_1(v)$ and $f_2(w)$, and does not easily generalize like the join and meet operations under minimum t-norm. Generally it is very difficult to obtain a closed-form expression for the result of the meet operation under product t-norm. One exception to this is for IT2 FSs, because for them $f_1(v) f_2(w) = 1$ when $f_1(v) = 1$ and $f_2(w) = 1$, and $f_1(v) f_2(w) = 0$ when $f_1(v) = 0$ or $f_2(w) = 0$.

Because of the complexity of computing the meet under the product t-norm, except for the next result and for IT2 FSs, it is not used in the rest of this book.

Theorem 7.7 (Sifting Theorem) *Consider the meet, under minimum or product t-norm, between a* type-2 fuzzy singleton $\tilde{A}$ *at* $x = x'$ *and a* GT2 FS $\tilde{B}$ *whose secondary MFs are normal. Then*

$$\tilde{A}(x) \sqcap \tilde{B}(x) = \begin{cases} \tilde{B}(x') & x = x' \\ 1/0 & x \neq x' \end{cases} \tag{7.17}$$

This is a very important theorem for Sect. 9.4.2.1, where an IT2 fuzzy system that is activated by singleton fuzzification is examined in great detail. Equation (7.17) indicates that for such a fuzzification, the input $x = x'$ to the fuzzy system sifts out the vertical slice of its antecedent MF $\tilde{B}(x')$, the result being a T1 FS.

Proof See Appendix 2, Sect. 2.5.

7.2.3 Complement of a GT2 FS

Theorem 7.8 *The* complement of GT2 FS $\tilde{A}$ *is computed in vertical-slice format, as* $(x \in X)$:

$$\bar{\tilde{A}} \Leftrightarrow \mu_{(\bar{\tilde{A}})_x}(u) \equiv \neg \mu_{\tilde{A}(x)} = \int_{u \in [0,1]} f_x(u)/(1-u) \quad x \in X \tag{7.18}$$

where $\neg$ *denotes the so-called* negation *operation* (Mizumoto and Tanaka 1976)[4].

Proof See Appendix 2, Sect. 2.6.

The use of the notation $\neg\mu_{\tilde{A}(x)}$ to indicate the negation of the type-1 secondary set $\tilde{A}(x)$ is yet another *shorthand notation*, but this time for the operations in the rightmost part of (7.18). It should be clear from (7.18) that $\neg\mu_{\tilde{A}(x)}$ is actually the MF of a T1 FS, namely $\mu_{\neg\mu_{\tilde{A}(x)}}(u)$.

What (7.18) says is that to perform the negation of the secondary set $\tilde{A}(x)$, $1-u$ must be computed at $u \in [0,1]$, and the secondary grade of $\mu_{\left(\bar{\tilde{A}}\right)_x}$ at $1-u$ is the corresponding secondary grade of $\mu_{\tilde{A}(x)}(u)$, $f_x(u)$. This must be done for $x \in X$ to obtain $\mu_{\bar{\tilde{A}}}$.

Example 7.12 This is a continuation of Example 7.1, in which $U = \{0, 0.1, \ldots, 0.9, 1\}$, $\mu_{\tilde{A}(x)}(u) = 0.5/0 + 0.7/0.1$, and $\mu_{\tilde{B}(x)}(u) = 0.3/0.4 + 0.9/0.8$. From (7.18), it follows that:

$$\begin{aligned} \mu_{\left(\bar{\tilde{A}}\right)_x}(u) &= \neg\mu_{\tilde{A}(x)} = 0.5/(1-0) + 0.7/(1-0.1) \\ &= 0.5/1 + 0.7/0.9 \end{aligned} \tag{7.19}$$

$$\begin{aligned} \mu_{\left(\bar{\tilde{B}}\right)}(u) &= \neg\mu_{\tilde{B}(x)} = 0.3/(1-0.4) + 0.9/(1-0.8) \\ &= 0.3/0.6 + 0.9/0.2 \end{aligned} \tag{7.20}$$

Theorem 7.9 (Mizumoto and Tanaka 1976) *If* $\tilde{A}(x)$ *is a normal convex T1 FS, then* $\neg\mu_{\tilde{A}(x)}$ *is also normal and convex.*

Proof That $\neg\mu_{\tilde{A}(x)}$ is normal and convex follows directly from (7.18).

Theorem 7.10 (Karnik and Mendel 2001) *If a T1 FS* F *has a MF* $f(\theta)$ $(\theta \in [0,1])$, *then* $\neg F$ *has a MF* $f(1-\theta)$.

Proof (Karnik and Mendel 2001) Substitute $\theta = 1-u$ into (7.18).

Example 7.13 Figure 7.8 shows an example of the complement of a Gaussian GT2 FS. The result in (b) was obtained by applying Theorem 7.10 to the secondary MFs in (a) for all $x \in [0,5]$. Again, observe that the principal MF of the result of the complement can be obtained by performing that operation on the principal MF of the participating GT2 FS. So, again, if one replaces the GT2 FS by a T1 FS whose

---

[4](Mizumoto and Tanaka 1976) express this as: $\bar{\tilde{A}} \Leftrightarrow \mu_{\bar{\tilde{A}}}(x) \equiv \neg\mu_{\tilde{A}}(x) = \int_u f_x(u)/(1-u)$. Following a similar line of reasoning as already given in Footnotes 2 and 3, $\neg\mu_{\tilde{A}}(x)$ has been used in (7.18) [rather than $\neg\tilde{A}(x)$] to preserve the link between Mizumoto and Tanaka's notation, since $\neg\mu_{\tilde{A}(x)}$ looks very similar to $\neg\mu_{\tilde{A}}(x)$.

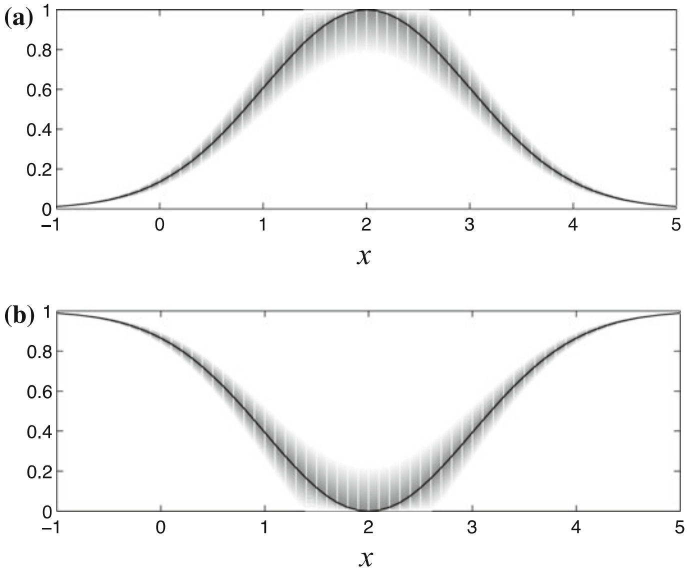


Fig. 7.8 Complement of a Gaussian GT2 FS. **a** Gaussian GT2 FS (heavy curve is the principal MF), and **b** its complement [heavy curve is the complement of the principal MF in (**a**)]

type-1 MF is the principal MF of the GT2 FS, all results reduce to the familiar ones for T1 FSs.

7.2.4 Remarks

7.2.4.1 Type-1 and Type-2 Set-Theoretic Operations Are Different

Secondary MFs are T1 FSs, and so some may wonder: is $\cup$ the same as $\sqcup$?; is $\cap$ the same as $\sqcap$?; and, is $-$ the same as $\neg$? This example, adapted from Mizumoto and Tanaka (1976), demonstrates that all three are different.

Example 7.14 Let $U = \{0, 0.1, \ldots, 0.9, 1\}$ and $\mu_{\tilde{A}(x)}$ and $\mu_{\tilde{B}(x)}$ be given as:

$$\mu_{\tilde{A}(x)}(u) = 0.5/0 + 0.7/0.1 + 0.3/0.2 \tag{7.21}$$

$$\mu_{\tilde{B}(x)}(u) = 0.9/0 + 0.6/0.1 + 0.2/0.2 \tag{7.22}$$

It follows from (7.3a) in which the minimum t-norm is used, that:

$$
\begin{aligned}
\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)} &= \frac{\min(0.5,0.9)}{0 \vee 0} + \frac{\min(0.5,0.6)}{0 \vee 0.1} + \frac{\min(0.5,0.2)}{0 \vee 0.2} + \frac{\min(0.7,0.9)}{0.1 \vee 0} + \frac{\min(0.7,0.6)}{0.1 \vee 0.1} \\
&\quad + \frac{\min(0.7,0.2)}{0.1 \vee 0.2} + \frac{\min(0.3,0.9)}{0.2 \vee 0} + \frac{\min(0.3,0.6)}{0.2 \vee 0.1} + \frac{\min(0.3,0.2)}{0.2 \vee 0.2} \\
&= 0.5/0 + 0.5/0.1 + 0.2/0.2 + 0.7/0.1 + 0.6/0.1 + 0.2/0.2 \\
&\quad + 0.3/0.2 + 0.3/0.2 + 0.2/0.2 \\
&= 0.5/0 + (0.5 \vee 0.7 \vee 0.6)/0.1 + (0.2 \vee 0.2 \vee 0.3 \vee 03 \vee 0.2)/0.2 \\
&= 0.5/0 + 0.7/0.1 + 0.3/0.2
\end{aligned}
\tag{7.23}
$$

Proceeding similarly for $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ from (7.7), and $\neg\mu_{\tilde{A}(x)}$ from (7.18), it follows that (Exercise 7.12):

$$\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)} = 0.7/0 + 0.6/0.1 + 0.2/0.2 \tag{7.24}$$

$$\neg\mu_{\tilde{A}(x)} = 0.5/1 + 0.7/0.9 + 0.3/0.8 \tag{7.25}$$

Using (2.17), (2.18), and (2.19) to compute $\mu_{\tilde{A}(x) \cup \tilde{B}(x)}$, $\mu_{\tilde{A}(x) \cap \tilde{B}(x)}$, and $\mu_{\overline{\tilde{A}(x)}}$, respectively, it follows that:

$$\mu_{\tilde{A}(x) \cup \tilde{B}(x)}(u) = 0.9/0 + 0.7/0.1 + 0.3/0.2 \tag{7.26}$$

$$\mu_{\tilde{A}(x) \cap \tilde{B}(x)}(u) = 0.5/0 + 0.6/0.1 + 0.2/0.2 \tag{7.27}$$

$$\mu_{\overline{\tilde{A}(x)}}(u) = 0.5/0 + 0.3/0.1 + 0.7/0.2 + 1/0.3 + 1/0.4 + \cdots + 1/1 \tag{7.28}$$

Consequently, $\tilde{A}(x) \cup \tilde{B}(x) \neq \mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$, $\tilde{A}(x) \cap \tilde{B}(x) \neq \mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$, and $\overline{\tilde{A}(x)} \neq \neg\mu_{\tilde{A}(x)}$.

7.2.4.2 Other Methods

The vertical-slice formulas for computing the join, meet, and negation, which are used to compute the union, intersection, and complement of GT2 FSs, are not the only way to compute these quantities. Coupland and John (2004, 2005, 2007, 2013) have shown how to compute the join and meet of $\tilde{A}$ and $\tilde{B}$ using methods from computational geometry (e.g., a modified Weiler–Atherton clipping algorithm, and a Bentley–Ottman plane sweep algorithm). Their approach is based on modeling a secondary MF geometrically as a (Coupland and John 2005): "set of connected straight line segments that need not be equally spaced across the domain," and is limited so far to the minimum t-norm and the maximum t-conorm. Based on extensive simulations of a two-rule fuzzy system in which each rule has two-antecedents, and the secondary MFs are discretized into 10 points, and are also described by two line segments, Coupland and John obtain over a "four and a half fold increase in inferencing speed".

Greenfield and John (2007) presented an optimized grid method for computing the join and meet, one that is much more computationally efficient than a direct approach to using (7.6) and (7.10).

A serious limitation of the geometric, grid, and even the vertical-slice methods is that they do not lead to closed-form formulas for the join and meet operations. Such formulas can be very useful when a T2 fuzzy system is designed by optimizing an objective function with respect to MF parameters and explicit derivatives of that function have to be computed so that optimal values of those parameters can be found.

It will be seen, in Sect. 7.3, that closed-form formulas for the join and meet operation are available for IT2 FSs, and in Sect. 7.4, that closed-form formulas are also available for the join and meet of GT2 FSs, when they are computed using horizontal slices.

7.2.4.3 Set-Theoretic Properties

Not only are the union, intersection, and complement performed with GT2 FSs, but sometimes, just as for T1 FSs, other important set-theoretic operations are performed on them using well-known laws; e.g., commutative, associative, distributive, and De Morgan's laws (see Tables 7.1 and 7.2 in Appendix 1 for a list of all the laws). An important question that needs to be answered is:

> Is it permissible to use a particular law for T2 FSs under maximum t-conorm and either minimum or product t-norms?

Our focus is just on the maximum t-conorm and the minimum or product t-norms, because, as was explained earlier for T1 FSs, these are the most widely used ones in the fuzzy system's literature. The question must, of course, be reexamined if one uses other t-conorms and t-norms. What complicates the study of the above laws for GT2 FSs is that their secondary MFs may in general be: normal and convex, or non-normal and convex, or normal and non-convex, or non-normal and non-convex.

Because the studies into the answers to this question, although important, are very technical, their details are presented in Appendix 1. Here, just some conclusions about them are summarized.

- All laws that are satisfied for T1 FSs for minimum and product t-norms are also satisfied for IT2 FSs; hence, all laws are satisfied for the minimum t-norm, but some are not satisfied for the product t-norm (see Table 2.8).
- For the *minimum t-norm*, all laws are satisfied for normal and convex secondary MFs; this provides a very strong reason for using such secondary MFs.

- For the *product t-norm*, the most number of laws (but not all of the laws) satisfied are for normal and convex secondary MFs, which again provides a very strong reason for using such secondary MFs.
- If a law is violated for normal and convex secondary MFs it is violated for the three other kinds of secondary MFs; however, if a law is not violated for normal and convex secondary MFs it may or may not be violated for the three other kinds of secondary MFs.

All of this means, therefore, that *for IT2 FSs, one must be as careful when using the maximum t-conorm and product t-norm as one was for T1 FSs, but for GT2 FSs one must be even more careful when using maximum t-conorm and product t-norm.* The design of a maximum t-conorm and product t-norm interval or general type-2 fuzzy system that involves the use of any of the violated laws (e.g., distributive and De Morgan's) will be in error. *Fortunately, one usually does not have to use any of the violated laws in the creation and design of a T2 fuzzy system*. The same cannot be said, in general, for other applications of interval or GT2 FSs.

7.2.4.4 Join and Meet Operations for T2 FSs with Non-convex Secondary MFs

Ruiz-Garcia et al. (2016) have extended Theorems 7.3 and 7.6 from convex to non-convex secondary MFs. They then apply those extensions to the more general kinds of non-closed IT2 FSs that were described in Sect. 6.6, some of which are illustrated in Example 6.18. It is left to the reader to explore these interesting but very early results.

7.3 Set-Theoretic Operations for IT2 FSs

This section focuses on set-theoretic operations for IT2 FSs because closed-form formulas can be obtained for the join, meet, and negation of such T2 FSs, which sets them apart from all other T2 FSs, and which is why, as of the year 2017, they have been the most widely used T2 FSs. Additionally, because of the horizontal-slice representation of a GT2 FS (Sect. 6.7.3)—which did not exist when the first edition of this book was published (Mendel 2001), the results from this section can also be used to compute the join, meet, and negation of GT2 FSs, as will be demonstrated in Sect. 7.4.

7.3.1 Union of IT2 FSs

Because the secondary grades of an IT2 FS are all equal to 1, the secondary MFs are *type-1 interval fuzzy numbers* (see Definition 2.6). Let F denote such a secondary MF, i.e., $\mu_F(u) = 1$ for $u \in [l, r]$.

Theorem 7.11 *Let* $F_1, \ldots,$ *and* F_n *be* n *type-1 interval fuzzy numbers having domains* $[l_1, r_1]$, …, *and* $[l_n, r_n]$, *respectively. The* join *of* $F_1, \ldots,$ and F_n, $\sqcup_{i=1}^{n} F_i$, *is also a type-1 interval fuzzy number with domain* $[(l_1 \vee l_2 \vee \cdots \vee l_n), (r_1 \vee r_2 \vee \cdots \vee r_n)]$, *where* $\vee$ *denotes maximum.*

Proof See Appendix 2, Sect. 2.7.

Corollary 7.3 *The* union *of two IT2 FSs*, $\tilde{A}$ *and* $\tilde{B}$, $\tilde{A} \cup \tilde{B}$, *is another IT2 FS, with* $\mathrm{FOU}(\tilde{A} \cup \tilde{B})$, *where* $(x \in X)$

$$\tilde{A} \cup \tilde{B} = 1/\mathrm{FOU}(\tilde{A} \cup \tilde{B}) = 1/[\underline{\mu}_{\tilde{A}}(x) \vee \underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{A}}(x) \vee \bar{\mu}_{\tilde{B}}(x)] \tag{7.29}$$

Proof This follows from Theorem 7.1, in which $f_x = g_x = 1$, and from Theorem 7.11 in which $F_1 = [l_1, r_1] = [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]$ and $F_2 = [l_2, r_2] = [\underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{B}}(x)]$.

The generalization of (7.29) to more that two IT2 FSs is straightforward, and follows from (7.29) and the associative property of T2 FSs, e.g.

$$\begin{aligned}\tilde{A} \cup \tilde{B} \cup \tilde{C} &= 1/\mathrm{FOU}(\tilde{A} \cup \tilde{B} \cup \tilde{C}) \\ &= 1/[\underline{\mu}_{\tilde{A}}(x) \vee \underline{\mu}_{\tilde{B}}(x) \vee \underline{\mu}_{\tilde{C}}(x), \bar{\mu}_{\tilde{A}}(x) \vee \bar{\mu}_{\tilde{B}}(x) \vee \bar{\mu}_{\tilde{C}}(x)] x \in X\end{aligned} \tag{7.30}$$

For discrete universes of discourse, replace the interval sets $[\bullet, \bullet]$ in (7.29) and (7.30) by the sets $\{\bullet, \ldots, \bullet\}$.

Corollary 7.3 puts into evidence that, for IT2 FSs, the computation of $\tilde{A} \cup \tilde{B}$ only involves using the type-1 lower and upper MFs of their respective FOUs. Observe, in (7.29), that $\underline{\mu}_{\tilde{A}}(x) \vee \underline{\mu}_{\tilde{B}}(x)$ and $\bar{\mu}_{\tilde{A}}(x) \vee \bar{\mu}_{\tilde{B}}(x)$ are in full agreement with the union of two T1 FSs as is given in (2.17) when $\vee$ in (7.29) is the maximum [or it can even be another t-conorm, in which case the maximum in (7.29) would be replaced by "s" (see Footnote 10 in Chap. 2)].

Corollary 7.3 also serves to bridge IT2 and IVFSs, because $\mathrm{FOU}(\tilde{A})$ and $\mathrm{FOU}(\tilde{B})$ are IVFSs (see the paragraph just below Definition 6.19). In the IVFS literature, one might see the union of IVFSs A and B stated as $(x \in X)$ $A \cup B = [\underline{A}(x) \vee \underline{B}(x), \bar{A}(x) \vee \bar{B}(x)]$, where interval arithmetic is used to obtain this result.

Example 7.15 Here is a totally different way to derive $\tilde{A} \cup \tilde{B}$ for IT2 FSs, one that does not need the concept of the join, and only uses type-1 mathematics [adapted from Mendel et al. (2006, pp. 812–813)]. From the Wavy-Slice Representation Theorem for IT2 FSs, Theorem 6.3, it follows that[5]:

$$\tilde{A} \cup \tilde{B} = \sum_{j=1}^{n_A} \tilde{A}_e^j \cup \sum_{i=1}^{n_B} \tilde{B}_e^i = \sum_{j=1}^{n_A} \sum_{i=1}^{n_B} \tilde{A}_e^j \cup \tilde{B}_e^i = 1/\mathrm{FOU}(\tilde{A} \cup \tilde{B}) \tag{7.31}$$

where n_A and n_B denote the number of embedded IT2 FSs that are associated with $\tilde{A}$ and $\tilde{B}$, respectively, and

$$\mathrm{FOU}(\tilde{A} \cup \tilde{B}) = \sum_{j=1}^{n_A} \sum_{i=1}^{n_B} A_e^j \cup B_e^i \tag{7.32}$$

in which A_e^j and B_e^i are embedded T1 FSs, and (using the maximum t-conorm)$(x \in X)$

$$A_e^j \cup B_e^i = \max\left\{\mu_{A_e^j}(x), \mu_{B_e^i}(x)\right\} \tag{7.33}$$

Consequently, (7.32) is a collection of $n_A \times n_B$ functions that contain a lower bounding function and an upper bounding function since both $\mu_{A_e^j}(x)$ and $\mu_{B_e^i}(x)$ are bounded for all values of x.

The upper and lower MFs of an IT2 FS are themselves embedded type-1 fuzzy sets. For $\tilde{A}$, $\bar{\mu}_{\tilde{A}}(x)$ and $\underline{\mu}_{\tilde{A}}(x)$ denote its upper and lower MFs, respectively, whereas for $\tilde{B}$, $\bar{\mu}_{\tilde{B}}(x)$ and $\underline{\mu}_{\tilde{B}}(x)$ denote its comparable quantities. It must therefore be true that $(x \in X)$:

$$\sup_{\forall j,i} \max\left\{\mu_{A_e^j}(x), \mu_{B_e^i}(x)\right\} = \max\left\{\bar{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{B}}(x)\right\} = \bar{\mu}_{\tilde{A}}(x) \vee \bar{\mu}_{\tilde{B}}(x) \tag{7.34}$$

$$\inf_{\forall j,i} \max\left\{\mu_{A_e^j}(x), \mu_{B_e^i}(x)\right\} = \max\left\{\underline{\mu}_{\tilde{A}}(x), \underline{\mu}_{\tilde{B}}(x)\right\} = \underline{\mu}_{\tilde{A}}(x) \vee \underline{\mu}_{\tilde{B}}(x) \tag{7.35}$$

From (7.31) to (7.35) one concludes that $(x \in X)$

$$\tilde{A} \cup \tilde{B} = 1\Big/ \sum_{j=1}^{n_A} \sum_{i=1}^{n_B} A_e^j \cup B_e^i = 1/[\underline{\mu}_{\tilde{A}}(x) \vee \underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{A}}(x) \vee \bar{\mu}_{\tilde{B}}(x)] \tag{7.36}$$

[5]Equation (7.31) involves summation and union signs. As in the type-1 case, where this mixed notation is used, the summation sign is simply shorthand for lots of + signs. The + indicates the union between members of a set, whereas the union sign represents the union of the sets themselves. Hence, by using both the summation and union signs, one is able to distinguish between the union of sets versus the union of members within a set.

which is [according to (7.3)] the join of $\tilde{A}(x)$ and $\tilde{B}(x)$, $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$. The right-hand side of (7.36) is in agreement with the results in Corollary 7.3; however, (7.36) has been derived entirely within the framework of Theorem 6.3 and wavy slices, and did not have to use any T2 FS mathematics to obtain it. The extension of (7.36) from two to n IT2 FSs is straightforward.

7.3.2 Intersection of IT2 FSs

Theorem 7.12 *Let* $F_1, \ldots,$ *and* F_n *be* n *type-1 interval fuzzy numbers having domains* $[l_1, r_1], \ldots,$ *and* $[l_n, r_n]$, *respectively. The* meet *of* $F_1, \ldots,$ and F_n, $\sqcap_{i=1}^{n} F_i$, *is also a type-1 interval fuzzy number with domain* $[(l_1 \bigstar l_2 \bigstar \cdots \bigstar l_n), (r_1 \bigstar r_2 \bigstar \cdots \bigstar r_n)]$, *where* $\bigstar$ *denotes either the minimum or product t-norms.*

Proof See Appendix 2, Sect. 2.8.

Corollary 7.4 *The intersection of two IT2 FSs,* $\tilde{A}$ *and* $\tilde{B}$, $\tilde{A} \cap \tilde{B}$, *is another IT2 FS, with* $\mathrm{FOU}(\tilde{A} \cap \tilde{B})$, *where* $(x \in X)$

$$\tilde{A} \cap \tilde{B} = 1/\mathrm{FOU}(\tilde{A} \cap \tilde{B}) = 1/[\underline{\mu}_{\tilde{A}}(x) \wedge \underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{A}}(x) \wedge \bar{\mu}_{\tilde{B}}(x)] \tag{7.37}$$

Proof This follows from Theorem 7.4, in which $f_x = g_x = 1$, and from Theorem 7.12 in which $F_1 = [l_1, r_1] = [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]$ and $F_2 = [l_2, r_2] = [\underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{B}}(x)]$.

The generalization of (7.37) to more that two IT2 FSs is straightforward, and follows from (7.37) and the associative property of T2 FSs, e.g., $(x \in X)$

$$\begin{aligned} \tilde{A} \cap \tilde{B} \cap \tilde{C} &= 1/\mathrm{FOU}(\tilde{A} \cap \tilde{B} \cap \tilde{C}) \\ &= 1/[\underline{\mu}_{\tilde{A}}(x) \wedge \underline{\mu}_{\tilde{B}}(x) \wedge \underline{\mu}_{\tilde{C}}(x), \bar{\mu}_{\tilde{A}}(x) \wedge \bar{\mu}_{\tilde{B}}(x) \wedge \bar{\mu}_{\tilde{C}}(x)] \end{aligned} \tag{7.38}$$

For discrete universes of discourse, replace the interval sets $[\bullet, \bullet]$ in (7.37) and (7.38) by the sets $\{\bullet, \ldots, \bullet\}$.

Corollary 7.4 puts into evidence that, for IT2 FSs, the computation of $\tilde{A} \cap \tilde{B}$ also only involves using the type-1 lower and upper MFs of their respective FOUs. Observe, in (7.37), that $\underline{\mu}_{\tilde{A}}(x) \wedge \underline{\mu}_{\tilde{B}}(x)$ and $\bar{\mu}_{\tilde{A}}(x) \wedge \bar{\mu}_{\tilde{B}}(x)$ are in full agreement with the intersection of two T1 FSs as is given in (2.18) when $\wedge$ in (7.37) is the minimum, and (2.23) when $\wedge$ in (7.37) is the product.

Corollary 7.4 also serves to bridge IT2 and IVFSs. In the IVFS literature, one might see the intersection of IVFSs A and B stated as $(x \in X)$ $A \cap B = [\underline{A}(x) \wedge \underline{B}(x), \bar{A}(x) \wedge \bar{B}(x)]$, where interval arithmetic is used to obtain this result.

Example 7.16 As in Example 7.15, here is a totally different way to derive $\tilde{A} \cap \tilde{B}$ for IT2 FSs, one that does not need the concept of the meet, and only uses type-1 mathematics (Mendel et al. 2006). From the Wavy-Slice Representation Theorem for IT2 FSs, Theorem 6.3, it follows that:

$$\tilde{A} \cap \tilde{B} = \sum_{j=1}^{n_A} \tilde{A}_e^j \cap \sum_{i=1}^{n_B} \tilde{B}_e^i = \sum_{j=1}^{n_A} \sum_{i=1}^{n_B} \tilde{A}_e^j \cap \tilde{B}_e^i = 1/\mathrm{FOU}(\tilde{A} \cap \tilde{B}) \tag{7.39}$$

where

$$\mathrm{FOU}(\tilde{A} \cap \tilde{B}) = \sum_{j=1}^{n_A} \sum_{i=1}^{n_B} A_e^j \cap B_e^i \tag{7.40}$$

and (using a t-norm) $(x \in X)$

$$A_e^j \cap B_e^i = \mu_{A_e^j}(x) \bigstar \mu_{B_e^i}(x) \tag{7.41}$$

Consequently, (7.40) is a collection of $n_A \times n_B$ functions that contain a lower bounding function and an upper bounding function since both $\mu_{A_e^j}(x)$ and $\mu_{B_e^i}(x)$ are bounded for all values of x.

It must therefore be true that $(x \in X)$:

$$\sup_{\forall j,i} \mu_{A_e^j}(x) \bigstar \mu_{B_e^i}(x) = \bar{\mu}_{\tilde{A}}(x) \bigstar \bar{\mu}_{\tilde{B}}(x) \tag{7.42}$$

$$\inf_{\forall j,i} \mu_{A_e^j}(x) \bigstar \mu_{B_e^i}(x) = \underline{\mu}_{\tilde{A}}(x) \bigstar \underline{\mu}_{\tilde{B}}(x) \tag{7.43}$$

From (7.39) to (7.43) one concludes that $(x \in X)$

$$\tilde{A} \cap \tilde{B} = 1/\sum_{j=1}^{n_A} \sum_{i=1}^{n_B} A_e^j \cap B_e^i = 1/[\underline{\mu}_{\tilde{A}}(x) \bigstar \underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{A}}(x) \bigstar \bar{\mu}_{\tilde{B}}(x)] \tag{7.44}$$

which is [according to (7.7)] the meet of $\tilde{A}(x)$ and $\tilde{B}(x)$, $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$. The right-hand side of (7.44) is in agreement with the results in Corollary 7.4; however, (7.44) has been derived entirely within the framework of Theorem 6.3 and wavy slices, and did not have to use any T2 FS mathematics to obtain it. The extension of (7.44) from two to n IT2 FSs is straightforward.

7.3.3 Complement of an IT2 FS

Theorem 7.13 *Let F be a type-1 interval fuzzy number having domain $[l, r]$. The* negation *of F, $\neg F$, is also a type-1 interval fuzzy number with domain $[1 - r, 1 - l]$.*

Proof This follows directly from Theorem 7.10.

Corollary 7.5 *The* complement *of IT2 FS* $\tilde{A}$, $\bar{\tilde{A}}$, *is another IT2 FS*, *with* $\mathrm{FOU}(\bar{\tilde{A}})$, i.e.

$$\bar{\tilde{A}} = 1/\mathrm{FOU}(\bar{\tilde{A}}) = 1/[1 - \bar{\mu}_{\tilde{A}}(x), 1 - \underline{\mu}_{\tilde{A}}(x)]x \in X \tag{7.45}$$

Proof This follows from Theorem 7.8, in which $f_x = 1$, and Theorem 7.13 in which $F = [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]$.

For discrete universes of discourse, replace the interval set $[\bullet, \bullet]$ in (7.45) by the set $\{\bullet, \ldots, \bullet\}$.

Corollary 7.5 again puts into evidence that, for IT2 FSs, the computation of $\bar{\tilde{A}}$ also only involves using the type-1 lower and upper MFs of $\mathrm{FOU}(\tilde{A})$. Observe, in (7.45), that $1 - \bar{\mu}_{\tilde{A}}(x)$ and $1 - \underline{\mu}_{\tilde{A}}(x)$ are in full agreement with the complement of a T1 FS that is given in (2.19), because when $\tilde{A} \to A$, $1 - \bar{\mu}_{\tilde{A}}(x) = 1 - \underline{\mu}_{\tilde{A}}(x) = 1 - \mu_A(x)$.

Corollary 7.5 again serves to bridge IT2 and IVFSs. In the IVFS literature, one might see the complement of IVFS A stated as $(x \in X)$ $\bar{A} = [1 - \bar{A}(x), 1 - \underline{A}(x))]$, where interval arithmetic is used to obtain this result.

Example 7.17 As in Examples 7.15 and 7.16, here is totally different way to derive $\bar{\tilde{A}}$ for an IT2 FS, one that does not need the concept of the negation, and only uses type-1 mathematics [adapted from Mendel et al. (2006, p. 813)]. Using Theorem 6.3, it follows that:

$$\bar{\tilde{A}} = \overline{\sum_{j=1}^{n_A} \tilde{A}_e^j} = \sum_{j=1}^{n_A} \bar{\tilde{A}}_e^j = 1/\mathrm{FOU}(\bar{\tilde{A}}) \tag{7.46}$$

where

$$\mathrm{FOU}(\bar{\tilde{A}}) = \sum_{j=1}^{n_A} \bar{A}_e^j \tag{7.47}$$

and $(x \in X)$

$$\bar{A}_e^j = 1 - \mu_{A_e^j}(x) \tag{7.48}$$

Consequently, (7.47) is a collection of n_A functions that contain a lower bounding function and an upper bounding function, since $\mu_{A_e^j}(x)$ are bounded for all values of x.

It must therefore be true that $(x \in X)$:

$$\sup_{\forall j} \left[1 - \mu_{A_e^j}(x_k)\right] = 1 - \underline{\mu}_{\tilde{A}}(x) \tag{7.49}$$

$$\inf_{\forall j}\left[1-\mu_{A_e^j}(x_k)\right]=1-\bar{\mu}_{\tilde{A}}(x) \tag{7.50}$$

In obtaining the right-hand parts of (7.49) and (7.50) use has been made of the facts that it is always true that $\bar{\mu}_{\tilde{A}}(x)\geq\underline{\mu}_{\tilde{A}}(x)$, consequently, it is always true that $1-\underline{\mu}_{\tilde{A}}(x)\geq 1-\bar{\mu}_{\tilde{A}}(x)$.

From (7.46) to (7.50), one concludes that $(x\in X)$:

$$\bar{\tilde{A}}=1\Big/\sum_{j=1}^{n_A}\bar{A}_e^j=1/[1-\bar{\mu}_{\tilde{A}}(x),1-\underline{\mu}_{\tilde{A}}(x)] \tag{7.51}$$

which is the negation of $\tilde{A}(x)$, $\neg\mu_{\tilde{A}(x)}$. The right-hand side of (7.51) is in agreement with the results in Corollary 7.5, however, (7.51) has been derived entirely within the framework of Theorem 6.3 and wavy slices, and did not have to use any T2 FS mathematics to obtain it.

7.4 Set-Theoretic Operations for GT2 FSs Computed by Using Horizontal Slices[6]

This section obtains different formulas for computing the union, intersection, and complement of closed[7] GT2 FSs (recall that *their secondary MFs are convex*), formulas that originate by expressing each type-1 secondary MF in terms of its α-cut representation as in (6.38), (6.40), and (6.41), i.e.

$$\tilde{A}=\int_{x\in X}\mu_{\tilde{A}(x)}(u)/x=\int_{x\in X}\left[\sup_{\alpha\in[0,1]}\alpha/\tilde{A}(x)_\alpha\right]/x=\int_{x\in X}\left[\sup_{\alpha\in[0,1]}\alpha/[a_\alpha(x),b_\alpha(x)]\right]/x \tag{7.52}$$

$$\tilde{B}=\int_{x\in X}\mu_{\tilde{B}(x)}(u)/x=\int_{x\in X}\left[\sup_{\alpha\in[0,1]}\alpha/\tilde{B}(x)_\alpha\right]/x=\int_{x\in X}\left[\sup_{\alpha\in[0,1]}\alpha/[a_\alpha(x),b_\alpha(x)]\right]/x \tag{7.53}$$

Once the derivations of union, intersection, and complement have been completed, using (7.52) and (7.53), which are in a vertical-slice format, they will be reorganized into a horizontal-slice format. The latter are very useful because in

[6]Readers who are only interested in IT2 FSs can skip this section as well as Sect. 7.5 and go directly to Sect. 7.6.

[7]Whether or not the results that are described in this section apply to other kinds of GT2 FSs remains to be studied.

Chap. 11 it will be shown that a GT2 fuzzy system can be expressed as the fuzzy union of a collection of α-level (horizontal-slice) IT2 fuzzy systems.

None of what is in this section appeared in the first edition of this book (Mendel 2001) because the horizontal-slice representation of a GT2 FS did not exist when that book was published.

7.4.1 Union of GT2 FSs

Theorem 7.14 *The* union *of the (closed) GT2 FSs* $\tilde{A}$ *and* $\tilde{B}$ *is computed in horizontal-slice format*, *as*:

$$\tilde{A} \cup \tilde{B} = \sup_{\alpha \in [0,1]} \alpha / \left\{ \int_{x \in X} [a_\alpha(x) \vee c_\alpha(x), b_\alpha(x) \vee d_\alpha(x)] \right\} = \bigcup_{\alpha \in [0,1]} \alpha / (\tilde{A}_\alpha \cup \tilde{B}_\alpha) \tag{7.54}$$

where $\vee$ *is the maximum*, *and* [*see* (6.47)]

$$\tilde{A}_\alpha = \int_{x \in X} [a_\alpha(x), b_\alpha(x)]/x \tag{7.55}$$

$$\tilde{B}_\alpha = \int_{x \in X} [c_\alpha(x), d_\alpha(x)]/x \tag{7.56}$$

Proof See Appendix 2, Sect. 2.9.

In addition to obtaining the horizontal-slice formula for $\tilde{A} \cup \tilde{B}$ in (7.54), a *new formula for the join* is easily obtained from that equation, i.e., $(x \in X)$:

$$\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)} = \sup_{\alpha \in [0,1]} [\alpha / \{[a_\alpha(x) \vee c_\alpha(x), b_\alpha(x) \vee d_\alpha(x)]\}] \tag{7.57}$$

This is a way to compute the T1 FS $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ using α-cuts that places into evidence the importance of α-cuts for GT2 FSs, and is arguably the simplest way to compute the join.

To compute $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ using (7.57), proceed as follows:

(1) Choose the number of α-cuts, k, for $\tilde{A}(x)$ and $\tilde{B}(x)$. Greater accuracy requires choosing a larger value of k.
(2) Compute

    a. $[a_{\alpha_j}(x), b_{\alpha_j}(x)]$ and $[c_{\alpha_j}(x), d_{\alpha_j}(x)]$, $j = 1, \ldots, k$

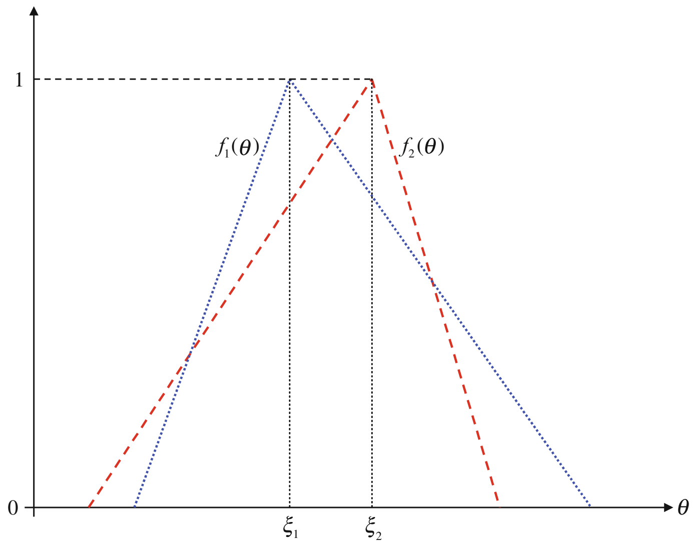


Fig. 7.9 Two overlapping convex type-1 secondary MFs (Mendel 2011; © 2011 IEEE)

b. $a_{\alpha_j}(x) \vee c_{\alpha_j}(x)$ and $b_{\alpha_j}(x) \vee d_{\alpha_j}(x)$, $j = 1, \ldots, k$
c. $\sup_{\alpha_j \in \{\alpha_1,\ldots,\alpha_k\}} \left[\alpha_j / \left\{\left[a_{\alpha_j}(x) \vee c_{\alpha_j}(x), b_{\alpha_j}(x) \vee d_{\alpha_j}(x)\right]\right\}\right]$ which provides $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$. Note that the α-cut of the join raised to level α is the bracketed term.

Example 7.18 (Mendel 2011) The purpose of this example is to show that the same results are obtained for the join of the two type-1 secondary MFs that are depicted in Fig. 7.9, using the vertical and horizontal-slice computations. Observe that these MFs intersect at three points.

To begin, the join of the two secondary MFs in Fig. 7.9 is computed using vertical-slice Eq. (7.5); the result is depicted as the heavy lines in Fig. 7.10. Next, $\mu_{F_1 \sqcup F_2}(\theta)$ is obtained using the α-cuts of $f_1(\theta)$ and $f_2(\theta)$ raised to level α_j, denoted $f_1(\theta|\alpha_j)$ and $f_2(\theta|\alpha_j)$, respectively. From Fig. 7.11:

- For $\alpha_0 = 0$, observe the following:

$$\begin{cases} f_1(\theta|0) = 0/[a_0, b_0] \\ f_2(\theta|0) = 0/[c_0, d_0] \end{cases} \tag{7.58}$$

$$\mu_{F_1 \sqcup F_2}(\theta|0) = 0/[\max(a_0, c_0), \max(b_0, d_0)] = 0/[a_0, b_0] \tag{7.59}$$

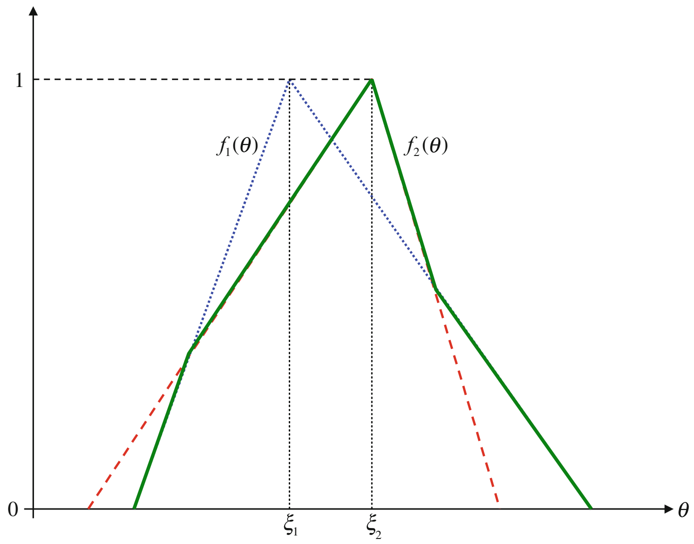


Fig. 7.10 Join (*in green*) of the two overlapping convex type-1 secondary MFs in Fig. 7.9 (Mendel 2011; © 2011 IEEE)

- For α_1, observe that:

$$\begin{cases} f_1(\theta|\alpha_1) = \alpha_1/[a_1, b_1] \\ f_2(\theta|\alpha_1) = \alpha_1/[c_1, d_1] \end{cases} \tag{7.60}$$

$$\mu_{F_1 \sqcup F_2}(\theta|\alpha_1) = \alpha_1/[\max(a_1, c_1), \max(b_1, d_1)] = \alpha_1/[a_1, b_1] \tag{7.61}$$

- For α_2, observe that:

$$\begin{cases} f_1(\theta|\alpha_2) = \alpha_2/[a_2, b_2] \\ f_2(\theta|\alpha_2) = \alpha_2/[c_2, d_2] \end{cases} \tag{7.62}$$

$$\mu_{F_1 \sqcup F_2}(\theta|\alpha_2) = \alpha_2/[\max(a_2, c_2), \max(b_2, d_2)] = \alpha_2/[c_2, b_2] \tag{7.63}$$

- Proceed in a similar manner for the remaining α_j that are depicted on Fig. 7.11.

Observe, in Fig. 7.11, that the results in (7.57) for horizontal-slice computations all lie on the green curve. By comparing the green curves in Figs. 7.11 and 7.10, one observes that they are the same. Q. E. D.

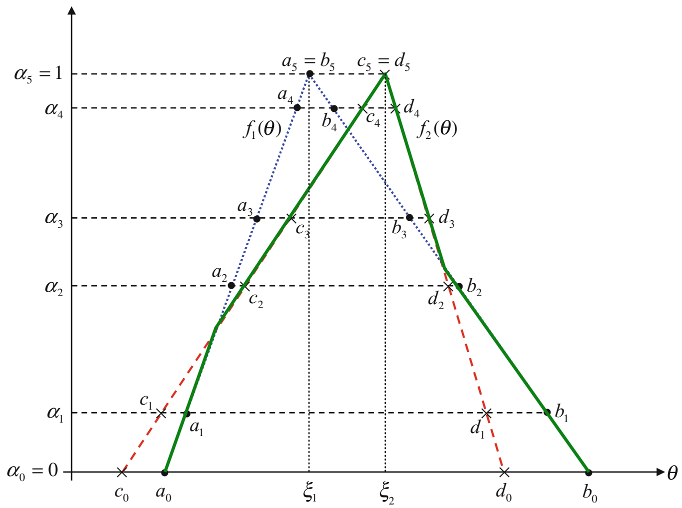


Fig. 7.11 Join (*in green*) of the two overlapping convex type-1 secondary MFs in Fig. 7.9 and some α-cuts raised to level α (Mendel 2011; © 2011 IEEE)

To compute the entire *horizontal slice*, $\tilde{A}_\alpha \cup \tilde{B}_\alpha$ [on the right-hand side of (7.54)], proceed as follows:

(1) Discretize X, calling these values of x, $x_1, \ldots, x_N$. Greater accuracy requires choosing a larger value of N.
(2) Compute (for $x_1, \ldots, x_N$):

a. $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$, the results of which are two bounded planes [as in (7.55) and (7.56)].
b. $\text{LMF}(\tilde{A}_\alpha) = a_\alpha(x_i)$, $\text{UMF}(\tilde{A}_\alpha) = b_\alpha(x_i)$, $\text{LMF}(\tilde{B}_\alpha) = c_\alpha(x_i)$, and $\text{UMF}(\tilde{B}_\alpha) = d_\alpha(x_i)$.
c. $\tilde{A}_\alpha \cup \tilde{B}_\alpha = [\text{LMF}(\tilde{A}_\alpha \cup \tilde{B}_\alpha), \text{UMF}(\tilde{A}_\alpha \cup \tilde{B}_\alpha)]$, in which $\text{LMF}(\tilde{A}_\alpha \cup \tilde{B}_\alpha) = a_\alpha(x_i) \vee c_\alpha(x_i)$ and $\text{UMF}(\tilde{A}_\alpha \cup \tilde{B}_\alpha) = b_\alpha(x_i) \vee d_\alpha(x_i)$.

Example 7.19 [Taken from Mendel et al. (2009, p. 1193)] Figure 7.12a depicts $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$, as well as their lower and upper MFs. Once the two α-planes are drawn, it is a relatively simple matter to draw the α-plane of $\tilde{A}_\alpha \cup \tilde{B}_\alpha$. Just take the maximum value of $\text{LMF}(\tilde{A}_\alpha)$ and $\text{LMF}(\tilde{B}_\alpha)$ to find $\text{LMF}(\tilde{A}_\alpha \cup \tilde{B}_\alpha)$. Similarly, take the maximum value of $\text{UMF}(\tilde{A}_\alpha)$ and $\text{UMF}(\tilde{B}_\alpha)$ to find $\text{UMF}(\tilde{A}_\alpha \cup \tilde{B}_\alpha)$. These lower and upper MFs, as well as the α-plane, $(\tilde{A} \cup \tilde{B})_\alpha = \tilde{A}_\alpha \cup \tilde{B}_\alpha$, are shown in Fig. 7.12b. The filled-in dots on that figure denote $\tilde{A}_\alpha \cup \tilde{B}_\alpha$ at x_1, x_2,…, and x_N.

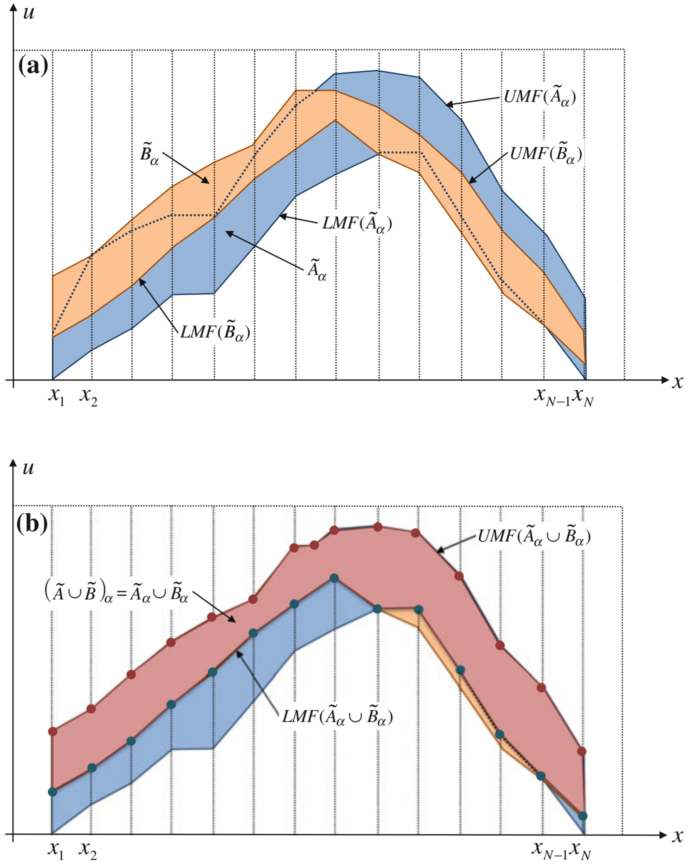


Fig. 7.12 **a** α-planes $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$, and **b** their union (Mendel et al. 2009; © 2009 IEEE)

7.4.2 Intersection of GT2 FSs

Theorem 7.15 *The* intersection *of the (closed) GT2 FSs* $\tilde{A}$ *and* $\tilde{B}$ *is computed in horizontal-slice format*, *as*:

$$\tilde{A} \cap \tilde{B} = \sup_{\alpha \in [0,1]} \alpha / \left\{ \int_{x \in X} [a_\alpha(x) \wedge c_\alpha(x), b_\alpha(x) \wedge d_\alpha(x)] \right\} = \bigcup_{\alpha \in [0,1]} \alpha / (\tilde{A}_\alpha \cap \tilde{B}_\alpha) \tag{7.64}$$

where $\wedge$ *is the minimum,*[8] *and* $\tilde{A}_\alpha$ *and* $\tilde{B}_\alpha$ *are in* (7.55) *and* (7.56), *respectively.*

Proof Because the derivation of (7.64) is so similar to the derivation of (7.54), it is left as an exercise for the reader (Exercise 7.26).

During the derivation of (7.64), one will obtain the following new formula for the meet $(x \in X)$:

$$\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)} = \sup_{\alpha\in[0,1]} [\alpha/\{[a_\alpha(x) \wedge c_\alpha(x), b_\alpha(x) \wedge d_\alpha(x)]\}] \tag{7.65}$$

Just as (7.57) is arguably the simplest way to compute the join, (7.65) is arguably the simplest way to compute the meet.

To compute $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ using (7.65), proceed as follows:

(1) Choose the number of α-cuts, k, for $\tilde{A}(x)$ and $\tilde{B}(x)$. Greater accuracy requires choosing a larger value of k.
(2) Compute
   a. $[a_{\alpha_j}(x), b_{\alpha_j}(x)]$ and $[c_{\alpha_j}(x), d_{\alpha_j}(x)]$, $j = 1, \ldots, k$
   b. $a_{\alpha_j}(x) \wedge c_{\alpha_j}(x)$ and $b_{\alpha_j}(x) \wedge d_{\alpha_j}(x)$, $j = 1, \ldots, k$
   d. $\sup_{\alpha_j\in\{\alpha_1,\ldots,\alpha_k\}} \left[\alpha_j/\left\{\left[a_{\alpha_j}(x) \wedge c_{\alpha_j}(x), b_{\alpha_j}(x) \wedge d_{\alpha_j}(x)\right]\right\}\right]$, which provides $\mu_{\tilde{A}(x_i)} \sqcap \mu_{\tilde{B}(x_i)}$. Note that the α-cut of the join raised to level α is the bracketed term.

Example 7.20 (Mendel 2011) This example is a continuation of Example 7.18. Its purpose is to show that the same results are obtained for the meet of the two type-1 secondary MFs that are depicted in Fig. 7.9, using the vertical and horizontal-slice computations.

To begin, the meet of the two secondary MFs in Fig. 7.9 is computed using vertical-slice Eq. (7.9); the result is depicted as the heavy lines in Fig. 7.13. Next, $\mu_{F_1 \sqcup F_2}(\theta)$ is obtained by using the α-cuts of $f_1(\theta)$ and $f_2(\theta)$ raised to level α_j, namely $f_1(\theta|\alpha_j)$ and $f_2(\theta|\alpha_j)$, respectively. $f_1(\theta|\alpha_j)$ and $f_2(\theta|\alpha_j)$, for j = 0, 1 and 2, are given in (7.58), (7.60), and (7.62), respectively. From Fig. 7.14:

- For $\alpha_0 = 0$, observe that:

$$\mu_{F\sqcap G}(\theta|0) = 0/[\min(a_0, c_0), \min(b_0, d_0)] = 0/[c_0, d_0] \tag{7.66}$$

- For α_1, observe that:

$$\mu_{F\sqcap G}(\theta|\alpha_1) = \alpha_1/[\min(a_1, c_1), \min(b_1, d_1)] = \alpha_1/[c_1, d_1] \tag{7.67}$$

[8]Whether or not $\wedge$ can also be the product is an open research question.

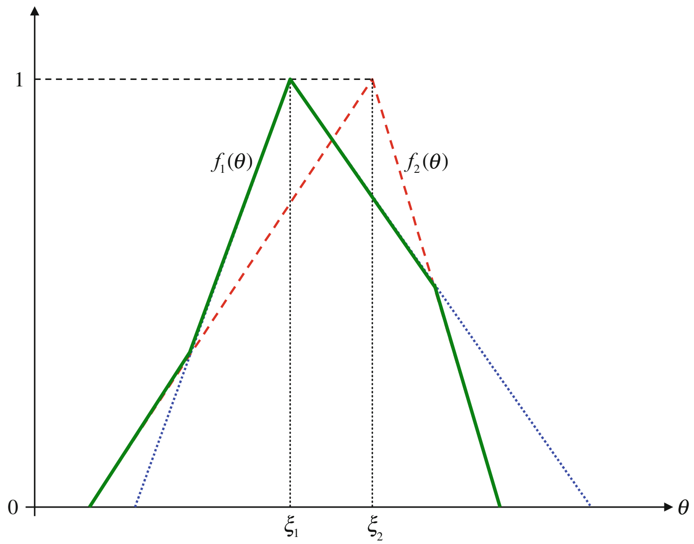


Fig. 7.13 Meet (*in green*) of the two overlapping convex type-1 secondary MFs in Fig. 7.9 (Mendel 2011; © 2011 IEEE)

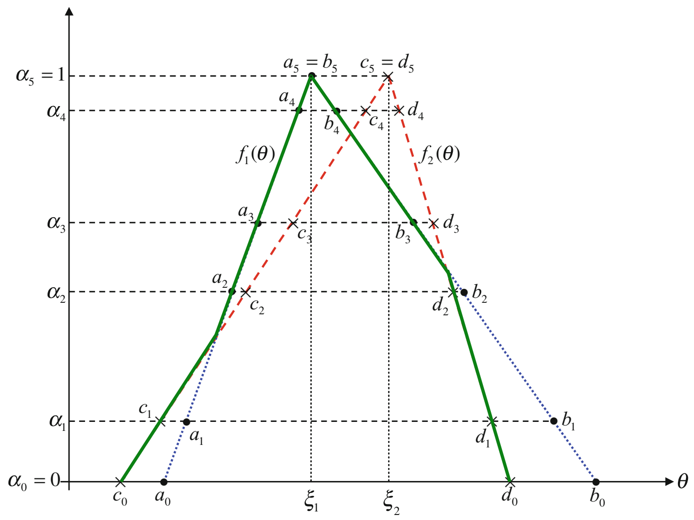


Fig. 7.14 Meet (*in green*) of the two overlapping convex type-1 secondary MFs in Fig. 7.9 and some α-cuts raised to level α (Mendel 2011; © 2011 IEEE)

- For α_2, observe that:

$$\mu_{F\sqcap G}(\theta|\alpha_2) = \alpha_2/[\min(a_2, c_2), \min(b_2, d_2)] = \alpha_2/[a_2, d_2] \tag{7.68}$$

- Proceed in a similar manner for the remaining α_j that are depicted on Fig. 7.14.

Observe, in Fig. 7.14, that the results in (7.65) for horizontal-slice computations all lie on the green curve. By comparing the green curves in Figs. 7.13 and 7.14, one observes that they are the same. Q. E. D.

To compute the entire *horizontal slice*, $\tilde{A}_\alpha \cap \tilde{B}_\alpha$ [on the right-hand side of (7.64)], proceed as follows:

(1) Discretize X, calling these values of x, $x_1, \ldots, x_N$. Greater accuracy requires choosing a larger value of N.
(2) Compute (for $x_1, \ldots, x_N$):

   a. $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$, the results of which are two bounded planes [as in (7.55) and (7.56)].
   b. $\text{LMF}(\tilde{A}_\alpha) = a_\alpha(x_i)$, $\text{UMF}(\tilde{A}_\alpha) = b_\alpha(x_i)$, $\text{LMF}(\tilde{B}_\alpha) = c_\alpha(x_i)$ and $\text{UMF}(\tilde{B}_\alpha) = d_\alpha(x_i)$.
   c. $\tilde{A}_\alpha \cap \tilde{B}_\alpha = [\text{LMF}(\tilde{A}_\alpha \cap \tilde{B}_\alpha), \text{UMF}(\tilde{A}_\alpha \cap \tilde{B}_\alpha)]$, in which $\text{LMF}(\tilde{A}_\alpha \cap \tilde{B}_\alpha) = a_\alpha(x_i) \wedge c_\alpha(x_i)$ and $\text{UMF}(\tilde{A}_\alpha \cap \tilde{B}_\alpha) = b_\alpha(x_i) \wedge d_\alpha(x_i)$.

Example 7.21 This example is a continuation of Example 7.19. Figure 7.12a depicts $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$, as well as their lower and upper MFs. Once the two α-planes are drawn, it is a relatively simple matter to draw the α-plane of $\tilde{A}_\alpha \cap \tilde{B}_\alpha$. Just take the minimum value of $\text{LMF}(\tilde{A}_\alpha)$ and $\text{LMF}(\tilde{B}_\alpha)$ to find $\text{LMF}(\tilde{A}_\alpha \cap \tilde{B}_\alpha)$. Similarly, take the minimum value of $\text{UMF}(\tilde{A}_\alpha)$ and $\text{UMF}(\tilde{B}_\alpha)$ to find $\text{UMF}(\tilde{A}_\alpha \cap \tilde{B}_\alpha)$. These lower and upper MFs, as well as the α-plane, $(\tilde{A} \cap \tilde{B})_\alpha = \tilde{A}_\alpha \cap \tilde{B}_\alpha$, are shown in Fig. 7.15. The filled-in dots on that figure denote $\tilde{A}_\alpha \cap \tilde{B}_\alpha$ at x_1, x_2,…, and x_N.

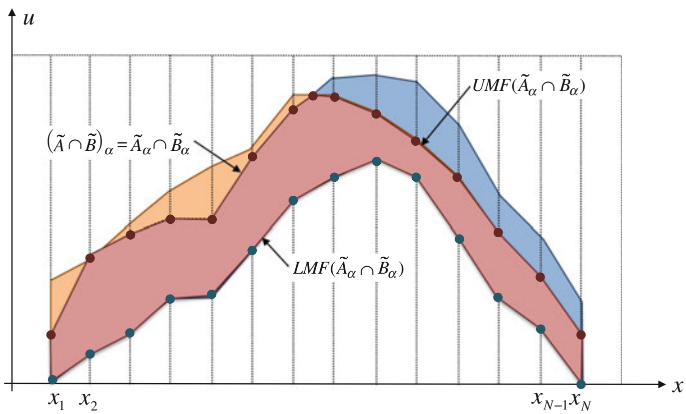


Fig. 7.15 The intersection of the Fig. 7.12a α-planes $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$

7.4.3 Complement of a GT2 FS

Theorem 7.16 *The* complement *of the GT2 FS* $\tilde{A}$ *is computed in horizontal-slice format, as*:

$$\bar{\tilde{A}} = \sup_{\alpha \in [0,1]} \alpha \Big/ \left\{ \int_{x \in X} [1 - b_\alpha(x), 1 - a_\alpha(x)] \right\} = \bigcup_{\alpha \in [0,1]} \alpha / \bar{\tilde{A}}_\alpha \tag{7.69}$$

where $\tilde{A}_\alpha$ *is in* (7.55).

Proof The derivation of (7.69) is also left to the reader (Exercise 7.27).

During the derivation of (7.69), one will obtain the following new formula for the negation $(x \in X)$:

$$\neg \mu_{\tilde{A}(x)} = \sup_{\alpha \in [0,1]} [\alpha / [1 - b_\alpha(x), 1 - a_\alpha(x)]] \tag{7.70}$$

Procedures for computing $\neg \mu_{\tilde{A}(x)}$ using (7.70) are also left to the reader (Exercise 7.28).

7.4.4 Historical Remarks

What seems not to be well known is that Zadeh (1975, pp. 242–247) proves that the intersection of two T2 FSs can be computed using the α-level sets of the secondary MFs (he does not call these MFs “secondary MFs” but, instead, refers to them as “fuzzy MFs”). This is summarized in his Eq. (3.104) but only for the minimum t-norm; hence, he was the first to recognize and show how set-theoretic operations for T2 FSs could be computed using α-cuts of the secondary MFs.

How the t-norm between two T2 FSs can be computed using α-cuts of the secondary MFs is also shown in Chen and Kawase (2000, Theorem 4.2). The specific connections between this work and Theorem 7.15 remain to be established (e.g., do the results in their Theorem 4.2 mean that (7.64) is also valid for the product t-norm?).

How to compute the intersection and union for two T2 FSs using what is comparable to the Wavy-Slice Representation theorem is shown in Tahayori et al. (2006, Theorems 1 and 2). A proof of their Theorem 1 is provided only for intersection.

How to compute set-theoretic operations using z-slices is given in Wagner and Hagras (2008, 2010, 2013). Their approach to proving Theorems 7.14 and 7.15 uses vertical slices and is very different from our proof that uses α-cuts.

7.5 Observations About Set Theory Computations

Stepping back from all of the details in Theorems 7.1–7.16, of which there are many, the following "big picture" has emerged:

- Set theory computations for IT2 FSs are two type-1 set theory computations, one involving type-1 LMFs, and the other involving type-1 UMFs
- Set theory computations for (closed) GT2 FSs can be expressed as comparable set theory computations for IT2 FSs of height α, which connects GT2 FSs and IT2 FSs very nicely
- Set theory computations for (closed) GT2 FSs make very heavy use of α-cuts of secondary MFs, so although α-cuts play no role in IT2 FS computations they play a central role in GT2 FS computations.

Whereas t-norms and t-conorms (see Footnotes 10 and 11 in Chap. 2) have been thoroughly studied for T1 FSs, the same cannot be said for GT2 FSs. Which of Theorems 7.1–7.16 are (or are not) valid for operations other than the maximum, minimum, and product remains to be studied. Whether this knowledge will have any bearing on T2 fuzzy systems is arguably debatable, because it is only the maximum, minimum, and product that have had significant bearings on T1 fuzzy systems.

7.6 Relations in General

The next few sections of this chapter examine type-2 fuzzy relations and their compositions that play an important role in T2 fuzzy systems.

Let $R(A_1,\ldots,A_n)$ denote a *relation* among the n non-fuzzy (crisp) sets $A_1,\ldots,A_n$. Then $R(A_1,\ldots,A_n)$ is a crisp subset of the Cartesian product $A_1\times\cdots\times A_n$ and $R(A_1,\ldots,A_n)\subset A_1\times\cdots\times A_n$. One can use the following MF to represent a non-fuzzy relation [see also (2.29)]:

$$\mu_R(a_1,\ldots,a_n)=\begin{cases}1, & \text{if } (a_1,\ldots,a_n)\in R(A_1,\ldots,A_n)\\ 0, & \text{otherwise}\end{cases} \tag{7.71}$$

A *binary relation* is the special case of a relation when $n = 2$ (see Sect. 2.5).

A *type-1 fuzzy relation* $R(A_1,\ldots,A_n)$ is (Lin and Lee 1996) a T1 FS that is defined on the Cartesian product space of crisp sets $A_1,\ldots,A_n$, where tuples $(a_1,a_2,\ldots,a_n)$ may have varying crisp degrees of membership $\mu_R(a_1,\ldots,a_n)$ within the relation. More specifically,

$$R(A_1,\ldots,A_n)=\int\limits_{A_1\times A_2\times\cdots\times A_n}\mu_R(a_1,a_2,\ldots,a_n)/(a_1,a_2,\ldots,a_n)\quad a_i\in A_i \tag{7.72}$$

In (7.72), at each $(a_1, a_2, \ldots, a_n)$ $\mu_R(a_1, a_2, \ldots, a_n)$ is a number $\in [0, 1]$. Binary type-1 fuzzy relations were described in Sect. 2.6.

A *type-2 fuzzy relation* $\tilde{R}(A_1, \ldots, A_n)$ is a T2 FS that is also defined on the Cartesian product space of crisp sets $A_1, \ldots, A_n$, where tuples $(a_1, a_2, \ldots, a_n)$ may have varying fuzzy degrees of membership $\mu_{\tilde{R}}(a_1, \ldots, a_n)$ within the relation. More specifically,

$$\tilde{R}(A_1, \ldots, A_n) = \int_{A_1 \times A_2 \times \cdots \times A_n} \mu_{\tilde{R}}(a_1, a_2, \ldots, a_n)/(a_1, a_2, \ldots, a_n) \quad a_i \in A_i \tag{7.73}$$

In (7.73), at each $(a_1, a_2, \ldots, a_n)\mu_{\tilde{R}}(a_1, \ldots, a_n)$ is a T1 FS [a secondary MF that is made more clear, e.g., in (7.75)].

John et al. (2006) note that: "Miyakoshi et al. (1980) use the term *fuzzy-fuzzy relation* to describe type-2 fuzzy relations and present an account of their properties and use them in a simple classification technique. Dubois and Prade (1980) refer to type-2 fuzzy relations as *fuzzy-valued fuzzy relations* and only discuss them briefly, while Klir and Folger (1988) mention type-2 fuzzy relations without actually defining them." They then contrast a type-2 fuzzy relation with crisp and type-1 relations in the following easy to understand way: "In a crisp relation, R $[R(U, V)]$ when $u_i \in U$ and $v_j \in V$ are related, then they are related with absolute certainty, so that $\mu_R(u_i, v_j) = 1$. In a type-1 fuzzy relation, when $u_i \in U$ and $v_j \in V$ are related, then they are related to a degree, so that $\mu_R(u_i, v_j) \in [0, 1]$. In a type-2 relation, $\tilde{R}$, when $u_i \in U$ and $v_j \in V$ are related, then they are related[9] *linguistically*, so that $\mu_{\tilde{R}}(u_i, v_j)$ is a type-1 fuzzy set." They also note: "… the type-2 fuzzy relation indicates a degree of membership which is itself a T1 FS—not a number in $[0, 1]$." The T1 FS elements of $\tilde{R}$ are analogous to the type-1 secondary MFs of a T2 FS.

All the previously discussed type-2 operations like join, meet, and negation can be used for type-2 relations.

Example 7.23 (Karnik and Mendel 1998c) As in Example 2.11, consider the type-1 fuzzy relation between real numbers u and v, namely "u is close to v." Recall, from that example, the MF chosen for this relation is

$$u_c(|u - v|) = \max\{(5 - |u - v|)/5, 0\} \tag{7.74}$$

This MF is depicted in Fig. 7.16a.

If one is not sure of the exact nature of the MF for this relation, perhaps because of the vagueness of the word "close," one could blur the MF in Fig. 7.16a to reflect this uncertainty. An example of this is depicted in Fig. 7.16b, where the FOU that is

[9]Today, I would be reluctant to use the word "linguistically," because, as has been explained in Chap. 5, words should be modeled using T2 FSs so as to account for their linguistic uncertainties. If T2 FSs are used as the entries in a fuzzy relation, then it becomes a type-3 fuzzy relation. To the best knowledge of this author, no research has been performed on such relations, e.g., how or if they can be used in a rule-based fuzzy system.

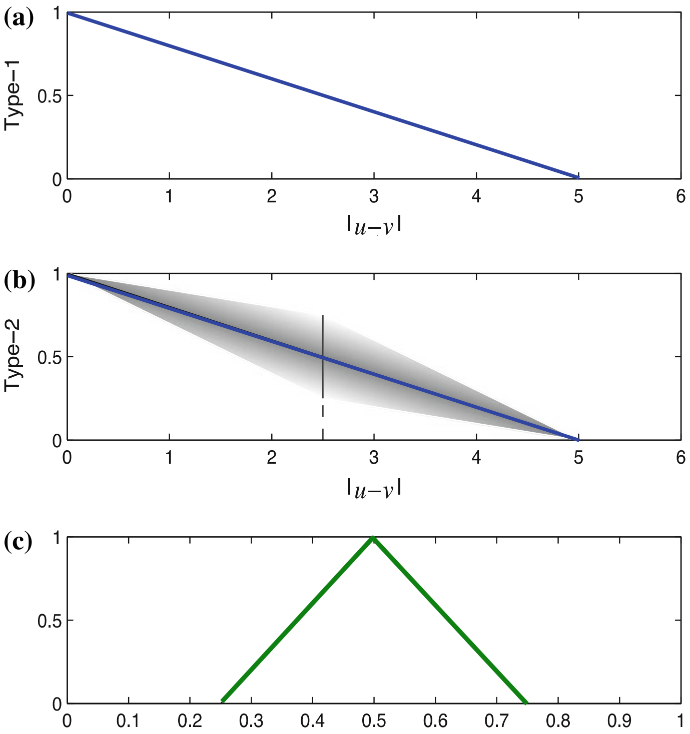


Fig. 7.16 Examples of MFs for **a** type-1 and **b** type-2 fuzzy relations. In (**b**) the *thick blue line* shows the primary memberships that have secondary grades equal to 1 (i.e., the principal MF). The intensity of color in the *gray area* is approximately proportional to the value of the secondary grades; *darker color* represents higher secondary grades. The primary membership corresponding to $|u - v| = 2.5$ is also indicated in the figure by the *dark vertical line*. **c** Shows the secondary MF corresponding to $|u - v| = 2.5$

associated with $u_{\tilde{c}}(|u-v|)$ is shown.[10] This FOU has been arbitrarily chosen so that when $|u-v|=0$ there is no additional fuzziness about "close." Similarly, when $|u-v|=5$, this is so far from $|u-v|=0$ that, again, there is no additional fuzziness about it. When $|u-v|=2.5$ the primary membership is the interval [0.25, 0.75], and the secondary grade reaches unity when the primary membership equals 0.5. Figure 7.16c shows an example of a triangular secondary MF corresponding to $|u-v|=2.5$. Other choices are possible for both the FOU and the

[10]For $0 \le |u-v| \le 2.5$, at each value of $|u-v|$, $u_{\tilde{c}}(|u-v|)$ is a T1 FS with primary membership arbitrarily chosen to be $[1-0.3|u-v|, 1-0.1|u-v|]$, and for $2.5 \le |u-v| \le 5$, at each value of $|u-v|$, $u_{\tilde{c}}(|u-v|)$ is a T1 FS with primary membership arbitrarily chosen to be $[0.5-0.1|u-v|, 1.5-0.3|u-v|]$.

shape of the secondary MFs, e.g., one could have used type-1 interval fuzzy numbers as the secondary MFs.

The coverage of type-2 relations and compositions in the next few sections is brief, because this is a book about rule-based fuzzy systems, and so our interest is only on those aspects of relations and compositions that will be used in Chaps. 9 and 11, which are about rule-based T2 fuzzy systems.

7.7 Type-2 Relations and Compositions on the Same Product Space

Consider two universes of discourse U and V. Let $\tilde{R}(U,V)$ and $\tilde{S}(U,V)$ be two type-2 fuzzy relations that are defined on the same product space. The elements of $\tilde{R}(U,V)$ and $\tilde{S}(U,V)$ are T1 FSs (i.e., secondary MFs), named $r_{(u,v)}$ and $s_{(u,v)}$, respectively. Using (7.73), $\tilde{R}(U,V)$ and $\tilde{S}(U,V)$ can be expressed as

$$\tilde{R}(U,V) = \int_{U\times V} \mu_{\tilde{R}}(u,v)/(u,v) = \int_{U\times V} \left[\int_{\xi_r\in[0,1]} \mu_{r_{(u,v)}}(\xi_r)/\xi_r\right]\Bigg/(u,v) \quad (7.75)$$

and

$$\tilde{S}(U,V) = \int_{U\times V} \mu_{\tilde{S}}(u,v)/(u,v) = \int_{U\times V} \left[\int_{\xi_s\in[0,1]} \mu_{s_{(u,v)}}(\xi_s)/\xi_s\right]\Bigg/(u,v) \quad (7.76)$$

The union and intersection of $\tilde{R}(U,V)$ and $\tilde{S}(U,V)$ can be expressed, as [see (7.3) and (7.7)] $[(u,v)\in U\times V]$:

$$\begin{aligned}\mu_{\tilde{R}\cup\tilde{S}}(u,v) &= \mu_{\tilde{R}}(u,v)\sqcup\mu_{\tilde{S}}(u,v)\\ &= \int_{\xi_r\in[0,1]}\int_{\xi_s\in[0,1]} \mu_{r_{(u,v)}}(\xi_r)\bigstar\mu_{s_{(u,v)}}(\xi_s)\Big/(\xi_r\vee\xi_s)\end{aligned} \quad (7.77)$$

and

$$\begin{aligned}\mu_{\tilde{R}\cap\tilde{S}}(u,v) &= \mu_{\tilde{R}}(u,v)\sqcap\mu_{\tilde{S}}(u,v)\\ &= \int_{\xi_r\in[0,1]}\int_{\xi_s\in[0,1]} \mu_{r_{(u,v)}}(\xi_r)\bigstar\mu_{s_{(u,v)}}(\xi_s)\Big/(\xi_r\wedge\xi_s)\end{aligned} \quad (7.78)$$

For discrete universes of discourse, replace the $\int$ by $\sum$, $\xi_r \in [0,1]$ by $\xi_r \in \Xi_r$, and $\xi_s \in [0,1]$ by $\xi_s \in \Xi_s$. Although these formulas look formidable, they are relatively easy to evaluate, as is demonstrated in the following example.[11]

Example 7.23 (Karnik and Mendel 1998c) As in Example 2.12, consider the two somewhat contradictory fuzzy relations "*u* is close to *v*" *and* "*u* is smaller than *v*," and also the less-contradictory relations "*u* is close to *v*" *or* "*u* is smaller than *v*." All relations are on the same product space $U \times V$. Recall that $U = \{u_1, u_2\} = \{2, 12\}$ and $V = \{v_1, v_2, v_3\} = \{1, 7, 13\}$. Here a type-2 version of these two relations is considered; i.e., it is assumed that there is some uncertainty about the type-1 membership grades. Let the secondary MFs for the type-2 case be as follows:

$$\mu_{\tilde{c}}(u,v) \equiv \begin{array}{c} \\ u_1 \\ u_2 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{pmatrix} 0.3/0.8+1/0.9+0.7/1 & 0.7/0.3+1/0.4+0.1/0.5 & 0.5/0+1/0.1 \\ 0.5/0+1/0.1 & 0.7/0.3+1/0.4+0.1/0.5 & 0.3/0.8+1/0.9+0.7/1 \end{pmatrix} \end{array} \tag{7.79}$$

and

$$\mu_{\tilde{s}}(u,v) \equiv \begin{array}{c} \\ u_1 \\ u_2 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{pmatrix} 1/0+0.9/0.1+0.4/0.5 & 0.8/0.3+0.8/0.4+0.9/0.5+1/0.6 & 0.9/0.9+1/1 \\ 1/0+0.1/0.1+0.1/0.2 & 1/0+0.3/0.1 & 1/0.3+0.9/0.4+0.4/0.5 \end{pmatrix} \end{array} \tag{7.80}$$

The secondary MFs in (7.79) and (7.80) have purposely been chosen so that the primary memberships that correspond to the membership grades in the type-1 case have unity secondary grades in the type-2 case (e.g., element 1/0.9 in $\mu_{\tilde{c}}(u_1, v_1)$, element 1/0.4 in $\mu_{\tilde{c}}(u_1, v_2)$, etc.). This can be interpreted as "perturbing" the membership matrices in (2.34) and (2.35) a little bit. Observe that pairs having the same membership grades in the type-1 case need not necessarily have the same complete secondary MFs in the type-2 case (e.g., see $\mu_{\tilde{s}}(u_1, v_1)$ and $\mu_{\tilde{s}}(u_2, v_1)$).

The secondary MFs for the union and intersection of these relations can be found from (7.77) and (7.78) as follows ($i = 1, 2$ and $j = 1, 2, 3$):

$$\begin{aligned} \mu_{\tilde{c} \cup \tilde{s}}(u_i, v_j) &= \mu_{\tilde{c}}(u_i, v_j) \sqcup \mu_{\tilde{s}}(u_i, v_j) \\ &= \sum_{\xi_r \in \Xi_r} \sum_{\xi_s \in \Xi_s} \mu_{r_{(u_i,v_j)}}(\xi_r) \bigstar \mu_{s_{(u_i,v_j)}}(\xi_s) \Big/ (\xi_r \vee \xi_s) \end{aligned} \tag{7.81}$$

and

$$\begin{aligned} \mu_{\tilde{c} \cap \tilde{s}}(u_i, v_j) &= \mu_{\tilde{c}}(u_i, v_j) \sqcap \mu_{\tilde{s}}(u_i, v_j) \\ &= \sum_{\xi_r \in \Xi_r} \sum_{\xi_s \in \Xi_s} \mu_{r_{(u_i,v_j)}}(\xi_r) \bigstar \mu_{s_{(u_i,v_j)}}(\xi_s) \Big/ (\xi_r \wedge \xi_s) \end{aligned} \tag{7.82}$$

[11] Although it should be possible to express (7.77) and (7.78) using α-cuts, this is not done because such formulas are not needed in the rest of this book.

Using (7.81) and (7.82) with minimum t-norm and maximum t-conorm, it is straightforward to show, for example, that:

$$\begin{aligned}\mu_{\tilde{c}\cup\tilde{s}}(u_1,v_1) &= (0.3/0.8+1/0.9+0.7/1)\sqcup(1/0+0.9/0.1+0.4/0.5)\\ &= 0.3/0.8+0.3/0.8+0.3/0.8+1/0.9+0.9/0.9\\ &\quad +0.4/0.9+0.7/1+0.7/1+0.4/1\\ &= 0.3/0.8+1/0.9+0.7/1\end{aligned} \tag{7.83}$$

and

$$\begin{aligned}\mu_{\tilde{c}\cap\tilde{s}}(u_1,v_1) &= (0.3/0.8+1/0.9+0.7/1)\sqcap(1/0+0.9/0.1+0.4/0.5)\\ &= 0.3/0+0.3/0.1+0.3/0.5+1/0+0.9/0.1\\ &\quad +0.4/0.5+0.7/0+0.7/0.1+0.4/0.5\\ &= 1/0+0.9/0.1+0.4/0.5\end{aligned} \tag{7.84}$$

The calculations of the remaining elements in $\mu_{\tilde{c}\cup\tilde{s}}(u,v)$ and $\mu_{\tilde{c}\cap\tilde{s}}(u,v)$ are left to the reader (Exercise 7.29). The final results are:

$$\mu_{\tilde{c}\cup\tilde{s}}(u,v)=\begin{array}{c}\\ u_1\\ u_2\end{array}\begin{array}{c}\begin{array}{ccc}v_1 & v_2 & v_3\end{array}\\ \begin{pmatrix}0.3/0.8+1/0.9+0.7/1 & 0.7/0.3+0.8/0.4+0.9/0.5+1/0.6 & 0.9/0.9+1/1\\ 0.5/0+1/0.1+0.1/0.2 & 0.7/0.3+1/0.4+0.1/0.5 & 0.3/0.8+1/0.9+0.7/1\end{pmatrix}\end{array} \tag{7.85}$$

and

$$\mu_{\tilde{c}\cap\tilde{s}}(u,v)=\begin{array}{c}\begin{array}{ccc}v_1 & v_2 & v_3\end{array}\\ \begin{pmatrix}1/0+0.9/0.1+0.4/0.5 & 0.8/0.3+1/0.4+0.1/0.5 & 0.5/0+1/0.1\\ 1/0+0.1/0.1 & 1/0+0.3/0.1 & 1/0.3+0.9/0.4+0.4/0.5\end{pmatrix}\end{array} \tag{7.86}$$

Comparing the results in (7.85) and (7.86) with those for the type-1 case, in (2.38) and (2.39), respectively, observe that:

1. Since the secondary MFs were chosen in such a way that the primary memberships corresponding to the membership grades in the type-1 case have unity secondary grades, the memberships of union and intersection also exhibit a similar structure; i.e., primary memberships for the union (intersection) of $\tilde{c}$ and $\tilde{s}$ that correspond to the membership grades for the union (intersection) in the type-1 case have unity secondary grades. For example, $\mu_{c\cap s}(u_1,v_2)=0.4$ and in $\mu_{\tilde{c}\cap\tilde{s}}(u_1,v_2)$ the secondary grade at the primary membership of 0.4 is 1.

2. The values of primary memberships of the union that have non-zero secondary grades are, in general, higher than those of the intersection of the aforementioned two relations, again indicating that the union (close *or* smaller) of the above two relations is treated with a higher degree of belief than their intersection (close *and* smaller).

7.8 Type-2 Relations and Compositions on Different Product Spaces

Next, the composition of type-2 fuzzy relations from different Cartesian product spaces that share a common set is considered, namely $\tilde{R}(U, V)$ and $\tilde{S}(V, W)$, e.g., u is *smaller* than v, and v is *close* to w. The composition of type-2 fuzzy relations from different Cartesian product spaces that share a common set is defined analogously to the crisp and type-1 fuzzy compositions, except that in the type-2 fuzzy case the sets are T2 FSs. Associated with type-2 fuzzy relation $\tilde{R}$ are its *T1 FS* elements $\mu_{\tilde{R}}(u, v)$, where $\mu_{\tilde{R}}(u, v) \in [0, 1]$; and, associated with type-2 fuzzy relation $\tilde{S}$ are its *T1 FS* elements $\mu_{\tilde{S}}(v, w)$, where $\mu_{\tilde{S}}(v, w) \in [0, 1]$.

Theorem 7.17 (Karnik and Mendel 2001, p. 337) *If*[12] *$\tilde{R}$ and $\tilde{S}$ are two type-2 fuzzy relations on $U \times V$ and $V \times W$, respectively, and $\mu_{\tilde{R}}(u, v)$ and $\mu_{\tilde{S}}(v, w)$ are normal T1 FSs*[13], *then the membership for any pair $(u, w), u \in U$ and $w \in W$, is non-zero if and only if there exists at least one $v \in V$ such that $\mu_{\tilde{R}}(u, v) \neq 1/0$ and $\mu_{\tilde{S}}(v, w) \neq 1/0$; this is equivalent to the following* extended sup-star composition[14] $(u \in U, w \in W)$:

$$\mu_{\tilde{R}\circ\tilde{S}}(u, w) = \sqcup_{v \in V}\left[\mu_{\tilde{R}}(u, v) \sqcap \mu_{\tilde{S}}(v, w)\right] \tag{7.87}$$

Proof Although the proof of this theorem is given in Appendix 2 (Sect. 2.10), when (7.87) is compared with (2.50), one observes that (7.87) is analogous to (2.50) where (as one might have anticipated) sup $\rightarrow \sqcup$ and $\bigstar \rightarrow \sqcap$.

Example 7.24 (Karnik and Mendel 1998c) As in Example 2.15, consider the relation "u is close to v" on $U \times V$, where $U = \{u_1, u_2\} = \{2, 12\}$ and $V = \{v_1, v_2, v_3\} = \{1, 7, 13\}$. Now consider another fuzzy relation "v is much bigger than w" on $V \times W$, where $W = \{w_1, w_2\} = \{4, 8\}$. Here type-2 versions of these two relations are obtained by adding some uncertainty to the type-1 relations in (2.34) and (2.51). $\mu_{\tilde{c}}(u, v)$ is given by (7.79), whereas $\mu_{\widetilde{mb}}(v, w)$ is:

[12] This theorem and its proof are taken from Karnik and Mendel (2001, pp. 337–338). Dubois and Prade (1979, 1980) gave a formula for the composition of type-2 relations, using the minimum t-norm as an extension of the type-1 sup-min composition. Their formula is the same as (7.87); however, Karnik and Mendel (1998c, 2001) have demonstrated the validity of (7.87) for product as well as minimum t-norms.

[13] As of the writing of this book, it is not known if this theorem is valid for non-normal type-1 fuzzy sets.

[14] Because (7.87) is in terms of the join and meet, it should be possible to obtain a version of it that uses α-cuts.

$$\mu_{\tilde{mb}}(v,w) \equiv \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{cc} \qquad\qquad w_1 \qquad\qquad & \qquad\qquad w_2 \qquad\qquad \end{array} \\ \begin{pmatrix} 1/0+0.6/0.1 & 1/0+0.1/0.1 \\ 0.4/0.5+1/0.6+0.9/0.7 & 1/0+0.8/0.1+0.2/0.2 \\ 0.7/0.9+1/1 & 0.5/0.6+1/0.7+0.7/0.8 \end{pmatrix} \end{array} \tag{7.88}$$

The composition of $\mu_{\tilde{c}}(u,v)$ and $\mu_{\tilde{mb}}(v,w)$ can be found using (7.87) as follows (i = 1, 2 and j = 1, 2, 3):

$$\begin{aligned} \mu_{\tilde{c}\circ\tilde{mb}}(u_i,w_j) = {} & \left[\mu_{\tilde{c}}(u_i,v_1) \sqcap \mu_{\tilde{mb}}(v_1,w_j)\right] \sqcup \left[\mu_{\tilde{c}}(u_i,v_2) \sqcap \mu_{\tilde{mb}}(v_2,w_j)\right] \\ & \sqcup \left[\mu_{\tilde{c}}(u_i,v_3) \sqcap \mu_{\tilde{mb}}(v_3,w_j)\right] \end{aligned} \tag{7.89}$$

For example (using (7.3) and (7.7) (in which $\int$ is replaced by $\sum$) and the minimum t-norm)

$$\begin{aligned} \mu_{\tilde{c}\circ\tilde{mb}}(u_1,w_1) &= [(0.3/0.8+1/0.9+0.7/1) \sqcap (1/0+0.6/0.1)] \\ &\quad \sqcup [(0.7/0.3+1/0.4+0.1/0.51) \sqcap (0.4/0.5+1/0.6+0.9/0.7)] \\ &\quad \sqcup [(0.5/0+1/0.1) \sqcap (0.7/0.9+1/1)] \\ &= 0.7/0.3+1/0.4+0.1/0.5 \end{aligned} \tag{7.90}$$

Doing all the calculations in a similar manner, one obtains (Exercise 7.31):

$$\mu_{\tilde{c}\circ\tilde{mb}}(u,w) = \begin{array}{c} \\ u_1 \\ u_2 \end{array} \begin{array}{c} \begin{array}{cc} \qquad\qquad w_1 \qquad\qquad & \qquad\qquad w_2 \qquad\qquad \end{array} \\ \begin{pmatrix} 0.7/0.3+1/0.4+0.1/0.5 & 0.5/0+1/0.1+0.2/0.2 \\ 0.3/0.8+1/0.9+0.7/1 & 0.5/0.6+1/0.7+0.7/0.8 \end{pmatrix} \end{array} \tag{7.91}$$

Observe that the results in (7.91) are quite similar to the results of the type-1 sup-star composition in (2.54); i.e., in the type-2 results, primary memberships corresponding to the memberships of the type-1 results have unity secondary grades; however, (7.91) provides additional uncertainty information to account for the uncertainties present in $\mu_{\tilde{c}}(u,v)$ and $\mu_{\tilde{mb}}(v,w)$.

7.9 Compositions of a T2 FS with a Type-2 Relation

Consider the case where one of the type-2 relations involved in the extended sup-star composition is just a T2 FS. The composition of a T2 FS $\tilde{R}(U)$ and a type-2 fuzzy relation $\tilde{S}(U,V)$ is given (Exercise 7.33) by the following special case of (7.87) Karnik and Mendel (1998c), (2001) $(v \in V)$:

$$\mu_{\tilde{R}\circ\tilde{S}}(v) = \sqcup_{u\in U}\left[\mu_{\tilde{R}(u)} \sqcap \mu_{\tilde{S}}(u,v)\right] \tag{7.92}$$

in which $\mu_{\tilde{R}(u)}$ is a secondary MF of $\tilde{R}$. This equation plays an important role as the inference mechanism for a rule whose antecedents (or consequent) are T2 FSs and *is the fundamental inference mechanism for rules in a T2 fuzzy system.*[15] It is the type-2 version of (2.56), where sup $\to \sqcup$ and $\bigstar \to \sqcap$.

Example 7.25 As in Example 2.16, consider again the relation "u is close to v" on $U \times V$, where $U = \{2, 12\}$ and $V = \{1, 7, 13\}$, however, here its type-2 version is used, where the secondary MF $\mu_{\tilde{c}}(u,v)$ is given in (7.79). Consider also the T2 FS "small" on U, whose secondary MF is obtained by adding some uncertainty to the MF of small, given in (2.57), as follows:

$$\mu_{\tilde{s}}(u) \equiv \begin{array}{cc} u_1 & u_2 \\ (0.5/0.7+1/0.9 & 1/0.1+0.3/0.4) \end{array} \tag{7.93}$$

The composition of the type-2 set "small" and the type-2 relation "u is close to v" can now be obtained using (7.92), as follows:

$$\mu_{\tilde{s}\circ\tilde{c}}(v_j) = \left[\mu_{\tilde{s}}(u_1) \sqcap \mu_{\tilde{c}}(u_1,v_j)\right] \sqcup \left[\mu_{\tilde{s}}(u_2) \sqcap \mu_{\tilde{c}}(u_2,v_j)\right] \tag{7.94}$$

where $j = 1, 2, 3$. Using (7.94) it is again straightforward to show that (Exercise 7.34)

$$\mu_{\tilde{s}\circ\tilde{c}}(v) = \begin{array}{ccc} v_1 & v_2 & v_3 \\ (0.5/0.7+0.3/0.8+1/0.9 & 0.7/0.3+1/0.4+0.1/0.5 & 1/0.1+0.3/0.4) \end{array} \tag{7.95}$$

Comparing (7.95) and (2.59), observe that the type-1 and type-2 results are again quite similar. In the type-2 results, primary memberships corresponding to the membership grades in the type-1 results have unity secondary grades.

When $\tilde{R}(U)$ is an IT2 FS, and all of the elements of $\tilde{S}(U,V)$ are type-1 interval fuzzy numbers (Definition 2.6), then the calculation of the meet and join in (7.92) can be performed by using Theorems 7.12 and 7.11, respectively.

[15]Because the join and meet can be computed using α-cuts, (7.92) can also be expressed in terms of α-cuts. That version of (7.92) is only used in Chap. 11, and so it is deferred until that chapter.

7.10 Type-2 Hedges[16]

Recall, from Sect. 2.9, that "a linguistic hedge or modifier is an operation that modifies the meaning of a term, or more generally, of a fuzzy set." When the fuzzy set for the linguistic term is a T2 FS (interval or general), then it seems reasonable that:

- For an IT2 FS:
  - *Concentration*: $\mathrm{con}(\tilde{F}) \equiv \left[\mathrm{LMF}\left(\mathrm{con}(\tilde{F})\right), \mathrm{UMF}\left(\mathrm{con}(\tilde{F})\right)\right]$, where:

$$\begin{cases} \mathrm{LMF}(\mathrm{con}(\tilde{F})) \equiv \left[\mathrm{LMF}(\tilde{F})\right]^2 \\ \mathrm{UMF}(\mathrm{con}(\tilde{F})) \equiv \left[\mathrm{UMF}(\tilde{F})\right]^2 \end{cases} \tag{7.96}$$

  - *Dilation*: $\mathrm{dil}(\tilde{F}) \equiv \left[\mathrm{LMF}\left(\mathrm{dil}(\tilde{F})\right), \mathrm{UMF}\left(\mathrm{dil}(\tilde{F})\right)\right]$, where:

$$\begin{cases} \mathrm{LMF}(\mathrm{dil}(\tilde{F})) \equiv \left[\mathrm{LMF}(\tilde{F})\right]^{1/2} \\ \mathrm{UMF}(\mathrm{dil}(\tilde{F})) \equiv \left[\mathrm{UMF}(\tilde{F})\right]^{1/2} \end{cases} \tag{7.97}$$

- For a GT2 FS[17] $(\alpha \in [0,1])$:
  - *Concentration*:

$$\mathrm{con}(\tilde{F}) \equiv \bigcup_{\alpha\in[0,1]} \alpha/\mathrm{con}(\tilde{F}_\alpha) = \bigcup_{\alpha\in[0,1]} \alpha/\left[\mathrm{LMF}\left(\mathrm{con}(\tilde{F}_\alpha)\right), \mathrm{UMF}\left(\mathrm{con}(\tilde{F}_\alpha)\right)\right] \tag{7.98}$$

  where

$$\begin{cases} \mathrm{LMF}(\mathrm{con}(\tilde{F}_\alpha)) \equiv \left[\mathrm{LMF}(\tilde{F}_\alpha)\right]^2 \\ \mathrm{UMF}(\mathrm{con}(\tilde{F}_\alpha)) \equiv \left[\mathrm{UMF}(\tilde{F}_\alpha)\right]^2 \end{cases} \tag{7.99}$$

---

[16]This is a speculative section in that the author could find no publications about type-2 hedges, and is therefore proposing formulas for them that seem reasonable.

[17]As of the year 2017, the only work that this author is familiar with about using GT2 FSs to model hedged words is Bilgin et al. (2012a, b, c), (2013a, b) that begin with a linguistic term modeled as either a left or right shoulder, apply n concentration hedges to it [the kind that make the membership of the hedged word contained within the membership of its less-hedged word (Sect. 2.9)], and then layer the hedged FOUs one on top of another at secondary grades equal to $1/n$. This *linear adjective* model is so far limited to shoulder models.

- *Dilation*:

$$\mathrm{dil}(\tilde{F}) \equiv \bigcup_{\alpha\in[0,1]} \alpha/\mathrm{dil}(\tilde{F}_\alpha) = \bigcup_{\alpha\in[0,1]} \alpha/\left[\mathrm{LMF}\left(\mathrm{dil}(\tilde{F}_\alpha)\right), \mathrm{UMF}\left(\mathrm{dil}(\tilde{F}_\alpha)\right)\right] \quad (7.100)$$

where

$$\begin{cases} \mathrm{LMF}(\mathrm{dil}(\tilde{F}_\alpha)) \equiv \left[\mathrm{LMF}(\tilde{F}_\alpha)\right]^{1/2} \\ \mathrm{UMF}(\mathrm{dil}(\tilde{F}_\alpha)) \equiv \left[\mathrm{UMF}(\tilde{F}_\alpha)\right]^{1/2} \end{cases} \quad (7.101)$$

It is important to remember (see Sect. 2.9) that: (1) The $\equiv$ sign has been used in these type-2 hedge MFs to convey the fact that their exponents are quite arbitrary; they can be changed depending upon one's interpretation of the hedges in question; and, (2) our advocated approach to hedges is to treat them as new linguistic terms, and not to use these contrived mathematical transformations.

7.11 Extension Principle for T2 FSs

The Extension Principle lets mathematical relationships between non-fuzzy variables be extended to fuzzy variables. The T1 FS Extension Principle (Sect. 2.10) was used in Sects. 7.2.1, 7.2.2, and 7.2.3 to derive the join, meet, and negation operations, respectively. In this section, Extension Principles are briefly described that apply directly to IT2 FSs or GT2 FSs. Our discussions are very brief because the results described next are not needed elsewhere in this book. They are included because there may be some future applications for rule-based systems that use T2 FSs where this material may be needed.

7.11.1 Extension Principle for IT2 FSs

Let f be a mapping from $X_1 \times \cdots \times X_r$ to a universe Y such that $y = f(x_1, \ldots, x_r) \in Y$, and $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_r$ be IT2 FSs in $X_1, X_2, \cdots, X_r$, respectively. Then, the extension of f to IT2 FSs, $\tilde{B} = f(\tilde{A}_1, \ldots, \tilde{A}_r)$, is determined by [e.g., Zeng et al. (2007), Hamrawi et al. (2010), Rajati and Mendel (2013)] $(y \in Y)$:

$$\mu_{\tilde{B}}(y) = 1/[\underline{\mu}_{\tilde{B}}(y), \bar{\mu}_{\tilde{B}}(y)] \quad (7.102)$$

$$\underline{\mu}_{\tilde{B}}(y) = \begin{cases} \sup\limits_{(x_1,\ldots,x_r)\in f^{-1}(y)} \min\{\underline{\mu}_{\tilde{A}_1}(x_1), \ldots, \underline{\mu}_{\tilde{A}_r}(x_r)\} \\ 0 \text{ if } f^{-1}(y) = \varnothing \end{cases} \quad (7.103)$$

$$\bar{\mu}_{\tilde{B}}(y) = \begin{cases} \sup\limits_{(x_1,\ldots,x_r)\in f^{-1}(y)} & \min\{\bar{\mu}_{\tilde{A}_1}(x_1),\ldots,\bar{\mu}_{\tilde{A}_r}(x_r)\} \\ & 0 \text{ if } f^{-1}(y) = \varnothing \end{cases} \tag{7.104}$$

Because an IT2 FS is completely described by its type-1 lower and upper MFs, to obtain the IT2 MF of $\tilde{B}$, the type-1 Extension Principle is applied twice, once to obtain its lower MF and once to obtain its upper MF.

Note that this is not a rigorous derivation of (7.102)–(7.104). Instead, it relies on what has already been learned in Examples 7.15–7.17 that operations on IT2 FSs lead to another IT2 FS, so the latter only requires the calculations of its lower and upper MFs. See, also Hamrawi et al. (2010, Theorem 3.2).

7.11.2 Extension Principle for GT2 FSs

Let f be a mapping from $X_1 \times \cdots \times X_r$ to a universe Y such that $y = f(x_1,\ldots,x_r) \in Y$, and $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_r$ be GT2 FSs in $X_1, X_2, \cdots, X_r$, respectively. Then, the extension of f to GT2 FSs, $\tilde{B} = f(\tilde{A}_1,\ldots,\tilde{A}_r)$, is determined by Rajati and Mendel (2013) ($y \in Y$):

$$\tilde{B} = \bigcup_{\alpha\in[0,1]} \alpha/[\underline{\mu}_{\tilde{B}_\alpha}(y), \bar{\mu}_{\tilde{B}_\alpha}(y)] \tag{7.105}$$

$$\underline{\mu}_{\tilde{B}_\alpha}(y) = \begin{cases} \sup\limits_{(x_1,\ldots,x_r)\in f^{-1}(y)} & \min\{\underline{\mu}_{\tilde{A}_{1\alpha}}(x_1),\ldots,\underline{\mu}_{\tilde{A}_{r\alpha}}(x_r)\} \\ & 0 \text{ if } f^{-1}(y) = \varnothing \end{cases} \tag{7.106}$$

$$\bar{\mu}_{\tilde{B}_\alpha}(y) = \begin{cases} \sup\limits_{(x_1,\ldots,x_r)\in f^{-1}(y)} & \min\{\bar{\mu}_{\tilde{A}_{1\alpha}}(x_1),\ldots,\bar{\mu}_{\tilde{A}_{r\alpha}}(x_r)\} \\ & 0 \text{ if } f^{-1}(y) = \varnothing \end{cases} \tag{7.107}$$

These equations are based on the horizontal-slice representation of a GT2 FS (Sect. 6.7.3). Equation (7.105) describes the α-plane $\tilde{B}_\alpha$ as an IT2 FS at level α using its LMF and UMF. The LMF of $\tilde{B}_\alpha$, $\underline{\mu}_{\tilde{B}_\alpha}(y)$, is computed using (7.106), which was obtained directly from (7.103) in which the LMFs of the α-planes, $\tilde{A}_{1\alpha},\ldots,\tilde{A}_{r\alpha}$, are used. Similarly, the UMF of $\tilde{B}_\alpha$, $\bar{\mu}_{\tilde{B}_\alpha}(y)$, is computed using (7.107), which was obtained directly from (7.104) in which the UMFs of the α-planes, $\tilde{A}_{1\alpha},\ldots,\tilde{A}_{r\alpha}$, are used.

Note that this is also not a rigorous derivation of (7.105)–(7.107). Instead, it relies on what has already been learned, namely that a GT2 FS can be represented as the fuzzy union of its α-planes raised to level α, where each of those α-planes raised to level α is an IT2 FS with membership grade α, and so one can then use (7.102)–(7.104) on each of those α-planes raised to level α. See, also Hamrawi et al. (2010), Theorem 4.5.

7.12 Functions of GT2 FSs Computed Using α-Planes

Recall, from Theorem 2.4 in Sect. 2.13, that α-cuts of a function of T1 FSs equal that function applied to the α-cuts of those T1 FSs. This α-cuts decomposition provides a very practical way to compute the MF of $y = f(x_1, \ldots, x_r) \in Y$ when $A_1, A_2, \ldots, A_r$ are T1 FSs in $X_1, X_2, \cdots, X_r$, respectively.

Because a GT2 FS can be represented as the fuzzy union of its α-planes raised to level α, and each α-plane is a union of α-cuts(over the primary variable), an analogous result is: α-planes *of a function of GT2 FSs should equal that function applied to the* α-planes *of those GT2 FSs*. This is a practical way to compute the MF of $y = f(x_1, \ldots, x_r) \in Y$ when $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_r$ are GT2 FSs in $X_1, X_2, \cdots, X_r$, respectively, and is a re-statement of (7.105)–(7.107).

This again is not a rigorously derived statement. Its proof is rather tedious because of simultaneously using α-cuts for vertical slices of secondary MFs and α-planes. See, e.g., Zhai and Mendel (2011), Kreinovich and Xiang (2008). This result, which has already been used in (7.105)–(7.107), is made use of also in Chap. 8, when the centroid of a GT2 FSs is computed, in some of the exercises for that chapter, and in Chap. 11.

7.13 Cartesian Product of T2 FSs

Let $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_n$ be T2 FSs in universes of discourse $X_1, X_2, \ldots, X_n$. The Cartesian product of $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_n$, $\tilde{A}_1 \times \tilde{A}_2 \times \cdots \times \tilde{A}_n$, is a T2 FS in the product space $X_1 \times X_2 \times \cdots \times X_n$ with the MF

$$\mu_{(\tilde{A}_1 \times \cdots \times \tilde{A}_n)(x_1, \ldots, x_n)} = \mu_{(\tilde{A}_1 \times \cdots \times \tilde{A}_n)(\mathbf{x})} = \mu_{\tilde{A}_1(x_1)} \sqcap \mu_{\tilde{A}_2(x_2)} \sqcap \cdots \sqcap \mu_{\tilde{A}_n(x_n)} \tag{7.108}$$

where $x_1 \in X_1, \ldots, x_n \in X_n$ and $\sqcap$ is the meet operation. In (7.108), $\mu_{\tilde{A}_i(x_i)}$ is the secondary MF of $\tilde{A}_i$ at x_i, and $\mu_{(\tilde{A}_1 \times \cdots \times \tilde{A}_n)(x_1, \ldots, x_n)}$ can also be viewed as a secondary MF, i.e., a T1 FS at $(x_1, \ldots, x_n)$.

Equation (7.108) should be compared with (2.116) to see that (7.108) can be obtained from (2.116) by letting $\bigstar \to \sqcap$.

Example 7.26 Consider two universes of discourse X_1 and X_2, where $X_1 = \{x_{11}, x_{12}, x_{13}\}$ and $X_2 = \{x_{21}, x_{22}\}$. Let $\tilde{F}$ and $\tilde{G}$ be T2 FSs on X_1 and X_2, respectively, where their secondary MFs are:

$$\begin{pmatrix} \mu_{\tilde{F}(x_{11})} \\ \mu_{\tilde{F}(x_{12})} \\ \mu_{\tilde{F}(x_{13})} \end{pmatrix} = \begin{pmatrix} 0.9/0.2 + 0.9/0.8 + 0.4/1 \\ 0.1/0.4 + 1/0.7 + 1/1 \\ 0.6/0 + 0.8/0.2 \end{pmatrix} \tag{7.109}$$

$$\begin{pmatrix} \mu_{\tilde{G}(x_{21})} \\ \mu_{\tilde{G}(x_{22})} \end{pmatrix} = \begin{pmatrix} 0.4/0.5 + 0.3/0.6 \\ 0.7/0.6 + 0.6/0.8 + 0.1/0.9 \end{pmatrix} \tag{7.110}$$

The MF of the Cartesian product of $\tilde{F}$ and $\tilde{G}$ can be obtained as $(i = 1, 2, 3 \text{ and } j = 1, 2)$

$$\mu_{(\tilde{F}\times\tilde{G})(x_{1i},x_{2j})} = \mu_{\tilde{F}(x_{1i})} \sqcap \mu_{\tilde{G}(x_{2j})} \tag{7.111}$$

It is left to the reader to show that (Exercise 7.36):

$$\mu_{(\tilde{F}\times\tilde{G})(x_1,x_2)} = \begin{array}{c} \\ x_{11} \\ x_{12} \\ x_{13} \end{array} \begin{array}{c} \begin{array}{cc} \qquad x_{21} \qquad\qquad & \qquad\qquad x_{22} \qquad\qquad \end{array} \\ \begin{pmatrix} 0.4/0.2 + 0.4/0.5 + 0.3/0.6 & 0.7/0.2 + 0.7/0.6 + 0.6/0.8 + 0.1/0.9 \\ 0.1/0.4 + 0.4/0.5 + 0.3/0.6 & 0.1/0.4 + 0.7/0.6 + 0.6/0.7 + 0.6/0.8 + 0.1/0.9 \\ 0.4/0 + 0.4/0.2 & 0.6/0 + 0.7/0.2 \end{pmatrix} \end{array} \tag{7.112}$$

7.14 Implications

Just as type-1 fuzzy systems use rules, so do type-2 fuzzy systems (much more will be said about this in Chaps. 9 and 11). The good news is that the structure of a rule does not change as one goes from a type-1 to a type-2 fuzzy system. All that changes is that (some or all of) the antecedents or consequents in a type-2 rule become T2 FSs instead of T1 FSs; hence, the structure of the lth type-2 Zadeh rule is:

$$\tilde{R}_Z^l \colon \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } x_2 \text{ is } \tilde{F}_2^l \text{ and} \ldots \text{and } x_p \text{ is } \tilde{F}_p^l \text{ THEN } y \text{ is } \tilde{G}^l \tag{7.113}$$

i.e.,

$$\tilde{R}_Z^l : \tilde{F}_1^l \times \cdots \times \tilde{F}_p^l \to \tilde{G}^l = \tilde{A}^l \to \tilde{G}^l \quad l = 1, \ldots, M \tag{7.114}$$

Recall from Sect. 3.4.1 that a rule is viewed as a *relation* between a collection of p antecedents and a single consequent. Here the type-1 implication MF, given in (3.7) and (3.8), is generalized to its type-2 counterpart, i.e.

$$\begin{aligned} \mu_{\tilde{R}_Z^l}(\mathbf{x}, y) &= \mu_{\tilde{A}^l \to \tilde{G}^l}(\mathbf{x}, y) = \mu_{(\tilde{F}_1^l \times \cdots \times \tilde{F}_p^l)(\mathbf{x})} \sqcap \mu_{\tilde{G}^l(y)} = \mu_{\tilde{F}_1^l(x_1)} \sqcap \cdots \sqcap \mu_{\tilde{F}_p^l(x_p)} \sqcap \mu_{\tilde{G}^l(y)} \\ &= \left[\sqcap_{i=1}^p \mu_{\tilde{F}_i^l(x_i)} \right] \sqcap \mu_{\tilde{G}^l(y)} \end{aligned} \tag{7.115}$$

where in this book the meet will either involve the minimum or the product t-norms. Comparing (7.115) and (3.8), observe that (7.115) is (3.8) when T and $\bigstar$ are replaced by $\sqcap$.

Just as (3.8) made use of the Cartesian product for $\mu_{F_1^l \times \cdots \times F_p^l}(\mathbf{x})$, given in (2.117), (7.115) makes use of the Cartesian product for $\mu_{(\tilde{F}_1^l \times \cdots \times \tilde{F}_p^l)(\mathbf{x})}$, given in (7.108).

For each $\mathbf{x}$ and y, $\mu_{R^l}(\mathbf{x}, y)$ in (7.115) is a T1 FS. To evaluate $\mu_{R^l}(\mathbf{x}, y)$ the results given in Sect. 7.2.2 for the meet need to be used. Equation (7.115) will be used extensively in Chaps. 9 and 11; its detailed calculations are deferred to those chapters.

Appendix 1: Properties of T2 FSs

This appendix is analogous to Appendix 1 in Chap. 2. It presents details about properties/laws of T2 FSs and, as in that Chap. 2 appendix, examines the following frequently used laws to see if they remain satisfied under maximum t-conorm and either minimum or product t-norms[18]:

> Reflexive, anti-symmetric, transitive, idempotent, commutative, associative, absorption, distributive, involution, De Morgan's, and identity

Recall, again, that the reason for doing this is that rules in a rule-based system may make use of the words "and", "or", "unless", "not", etc., but all of the mathematics for such a system is worked out in this book only for canonical rules that use the words "and" and "or". Section 3.2 showed how the former rules are transformed into the canonical rules by using some of the above laws. So, it is important to know when or if the use of these laws is correct when rules use T2 FSs.

Because the union and intersection of T2 FSs require computing the join and meet of the secondary MFs of these sets, set-theoretic laws for T2 FSs can be expressed in terms of secondary MFs.

Tables 7.1 and 7.2 are the type-2 counterparts to Table 2.8. Table 7.1 is for the minimum t-norm, whereas Table 7.2 is for the product t-norm. What complicates the study of the above laws for GT2 FSs is that their secondary MFs may in general be: normal and convex, or non-normal and convex, or normal and non-convex, or non-normal and non-convex, which is why Tables 7.1 and 7.2 have four columns with those headings.

For the product t-norm there can be two different situations, namely: (1) $\bigstar$ = product and $\wedge$ = product, and (2) $\bigstar$ = product and $\wedge$ = minimum. Only Situation 1 is considered in this book because it seems more consistent to this author to choose both $\bigstar$ and $\wedge$ to be the same. Situation 2 was introduced in

[18]Walker and Walker (2005, 2006, 2009, 2014) and Harding et al. (2010) have many interesting and important mathematical results about the join and meet that are beyond the scope of this book.

Table 7.1 Summary of set-theoretic laws and whether or not they are satisfied for GT2 FSs under *maximum t-conorm* and *minimum t-norm*[a]

Set-theoretic laws		Normal/convex[b]	Non-normal/convex	Normal/non-convex	Non-normal/non-convex[c]
Reflexive	$\mu_{\tilde{A}} \subseteq \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
Anti-symmetric	$\mu_{\tilde{A}} \subseteq \mu_{\tilde{B}}, \mu_{\tilde{B}} \subseteq \mu_{\tilde{A}} \Rightarrow \mu_{\tilde{A}} = \mu_{\tilde{B}}$	Yes	Yes	Yes	Yes
Transitive	$\mu_{\tilde{A}} \subseteq \mu_{\tilde{B}}, \mu_{\tilde{B}} \subseteq \mu_{\tilde{C}} \Rightarrow \mu_{\tilde{A}} \subseteq \mu_{\tilde{C}}$	Yes	Yes	Yes	Yes
Idempotent	$\mu_{\tilde{A}} \sqcup \mu_{\tilde{A}} = \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
	$\mu_{\tilde{A}} \sqcap \mu_{\tilde{A}} = \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
Commutative	$\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}} = \mu_{\tilde{B}} \sqcup \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
	$\mu_{\tilde{A}} \sqcap \mu_{\tilde{B}} = \mu_{\tilde{B}} \sqcap \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
Associative	$(\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}) \sqcup \mu_{\tilde{C}} = \mu_{\tilde{A}} \sqcup (\mu_{\tilde{B}} \sqcup \mu_{\tilde{C}})$	Yes	Yes	Yes	Yes
	$(\mu_{\tilde{A}} \sqcap \mu_{\tilde{B}}) \sqcap \mu_{\tilde{C}} = \mu_{\tilde{A}} \sqcap (\mu_{\tilde{B}} \sqcap \mu_{\tilde{C}})$	Yes	Yes	Yes	Yes
Absorption	$\mu_{\tilde{A}} \sqcap (\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}) = \mu_{\tilde{A}}$	Yes	**No**	**No**	**No**
	$\mu_{\tilde{A}} \sqcup (\mu_{\tilde{A}} \sqcap \mu_{\tilde{B}}) = \mu_{\tilde{A}}$	Yes	**No**	**No**	**No**
Distributive	$\mu_{\tilde{A}} \sqcap (\mu_{\tilde{B}} \sqcup \mu_{\tilde{C}}) = (\mu_{\tilde{A}} \sqcap \mu_{\tilde{B}}) \sqcup (\mu_{\tilde{A}} \sqcap \mu_{\tilde{C}})$	Yes	Yes	**No**	**No**
	$\mu_{\tilde{A}} \sqcup (\mu_{\tilde{B}} \sqcap \mu_{\tilde{C}}) = (\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}) \sqcap (\mu_{\tilde{A}} \sqcup \mu_{\tilde{C}})$	Yes	Yes	**No**	**No**
Involution	$\mu_{\bar{\bar{\tilde{A}}}} = \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
De Morgan's laws	$\overline{\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}} = \mu_{\bar{\tilde{A}}} \sqcap \mu_{\bar{\tilde{B}}}$	Yes	Yes	Yes	Yes
	$\overline{\mu_{\tilde{A}} \sqcap \mu_{\tilde{B}}} = \mu_{\bar{\tilde{A}}} \sqcup \mu_{\bar{\tilde{B}}}$	Yes	Yes	Yes	Yes
Identity	$\mu_{\tilde{A}} \sqcup (1/0) = \mu_{\tilde{A}}$	Yes	**No**	Yes	**No**
	$\mu_{\tilde{A}} \sqcup (1/1) = \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
	$\mu_{\tilde{A}} \sqcup (1/1) = 1/1$	Yes	Yes	Yes	Yes
	$\mu_{\tilde{A}} \sqcup (1/0) = 1/0$	Yes	**No**	Yes	**No**

[Adapted from Mizumoto and Tanaka (1976, Table II) and Karnik and Mendel (2001, Table 1)]. In this table, $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ are short for $\tilde{A}(x)$, $\tilde{B}(x)$, and $\tilde{C}(x)$, respectively, and the properties that are listed in the last four columns are for the secondary MFs

[a]Each of the table's laws applies at a specific value of x; but, for notational convenience, the secondary MFs are not shown as functions of x, i.e., $\mu_{\tilde{A}}$ is short for $\mu_{\tilde{A}(x)}$

[b]Table 8.2 in Mendel (2001) was only for this situation

[c]Mizumoto and Tanaka (1976) call this "fuzzy grades"

Table 7.2 Summary of set-theoretic laws and whether or not they are satisfied for GT2 FSs under *maximum t-conorm* and *product t-norm*[a]

Set-theoretic laws		Normal/convex[b,c]	Non-normal/convex	Normal/non-convex	Non-normal/non-convex[d]
Reflexive	$\mu_{\tilde{A}} \subseteq \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
Anti-symmetric	$\mu_{\tilde{A}} \subseteq \mu_{\tilde{B}}, \mu_{\tilde{B}} \subseteq \mu_{\tilde{A}} \Rightarrow \mu_{\tilde{A}} = \mu_{\tilde{B}}$	Yes	Yes	Yes	Yes
Transitive	$\mu_{\tilde{A}} \subseteq \mu_{\tilde{B}}, \mu_{\tilde{B}} \subseteq \mu_{\tilde{C}} \Rightarrow \mu_{\tilde{A}} \subseteq \mu_{\tilde{C}}$	Yes	Yes	Yes	Yes
Idempotent	$\mu_{\tilde{A}} \sqcup \mu_{\tilde{A}} = \mu_{\tilde{A}}$	**No**	**No**	**No**	**No**
	$\mu_{\tilde{A}} \sqcup \mu_{\tilde{A}} = \mu_{\tilde{A}}$	**No**	**No**	**No**	**No**
Commutative	$\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}} = \mu_{\tilde{B}} \sqcup \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
	$\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}} = \mu_{\tilde{B}} \sqcup \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
Associative	$(\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}) \sqcup \mu_{\tilde{C}} = \mu_{\tilde{A}} \sqcup (\mu_{\tilde{B}} \sqcup \mu_{\tilde{C}})$	Yes	Yes	Yes	Yes
	$(\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}) \sqcup \mu_{\tilde{C}} = \mu_{\tilde{A}} \sqcup (\mu_{\tilde{B}} \sqcup \mu_{\tilde{C}})$	Yes	Yes	Yes	Yes
Absorption	$\mu_{\tilde{A}} \sqcup (\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}) = \mu_{\tilde{A}}$	**No**	**No**	**No**	**No**
	$\mu_{\tilde{A}} \sqcup (\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}) = \mu_{\tilde{A}}$	**No**	**No**	**No**	**No**
Distributive	$\mu_{\tilde{A}} \sqcup (\mu_{\tilde{B}} \sqcup \mu_{\tilde{C}}) = (\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}) \sqcup (\mu_{\tilde{A}} \sqcup \mu_{\tilde{C}})$	**No**	**No**	**No**	**No**
	$\mu_{\tilde{A}} \sqcup (\mu_{\tilde{B}} \sqcup \mu_{\tilde{C}}) = (\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}) \sqcup (\mu_{\tilde{A}} \sqcup \mu_{\tilde{C}})$	**No**	**No**	**No**	**No**
Involution	$\mu_{\bar{\bar{\tilde{A}}}} = \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
De Morgan's Laws	$\overline{\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}} = \mu_{\bar{\tilde{A}}} \sqcup \mu_{\bar{\tilde{B}}}$	Yes	Yes	Yes	Yes
	$\overline{\mu_{\tilde{A}} \sqcup \mu_{\tilde{B}}} = \mu_{\bar{\tilde{A}}} \sqcup \mu_{\bar{\tilde{B}}}$	Yes	Yes	Yes	Yes
Identity	$\mu_{\tilde{A}} \sqcup (1/0) = \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
	$\mu_{\tilde{A}} \sqcup (1/1) = \mu_{\tilde{A}}$	Yes	Yes	Yes	Yes
	$\mu_{\tilde{A}} \sqcup (1/1) = 1/1$	Yes	**No**	Yes	**No**
	$\mu_{\tilde{A}} sqcap(1/0) = 1/0$	Yes	**No**	Yes	**No**

[Adapted from Karnik and Mendel (2001), Table 1]. In this table, $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ are short for $\tilde{A}(x)$, $\tilde{B}(x)$, and $\tilde{C}(x)$, respectively, and the properties that are listed in the last four columns are for the secondary MFs

[a]Each of the table's laws applies at a specific value of x; but, for notational convenience, the secondary MFs are not shown as functions of x; i.e., $\mu_{\tilde{A}}$ is short for $\mu_{\tilde{A}}(x)$

[b]Table 8.2 in Mendel (2001) was only for this situation

[c]For IT2 FSs the entries in the "Normal/Convex" column of this table change to the ones in the "Product t-norm" column of Table 2.8 (see Exercise 7.39)

[d]Mizumoto and Tanaka (1976) call this "fuzzy grades"

Mizumoto and Tanaka (1976) and set-theoretic properties for it can be found in that reference.

The secondary MFs for an IT2 FS are type-1 interval fuzzy numbers, and so they are normal and convex. Consequently for IT2 FSs, one only needs to examine the "Normal/Convex" columns of Tables 7.1 and 7.2. *All of the above laws/properties that are satisfied for T1 FSs for minimum and product t-norms are also satisfied for IT2 FSs* (see Exercises 7.37 and 7.38).

Observe, from Tables 7.1 and 7.2, that:

- For the minimum t-norm (Table 7.1), all laws are satisfied for normal and convex secondary MFs; this provides a very strong reason for using such secondary MFs.
- For the product t-norm (Table 7.2), the most number of laws (but not all of the laws) satisfied are for normal and convex secondary MFs, which again provides a very strong reason for using such secondary MFs.
- If a law is violated for normal and convex secondary MFs (Table 7.2) it is violated for the three other kinds of secondary MFs; however, if a law is not violated for normal and convex secondary MFs (Tables 7.1 and 7.2), it may or may not be violated for the three other kinds of secondary MFs.

Recall that all operations on GT2 FSs sets collapse to their type-1 counterparts when each secondary MF is replaced by its support at which the secondary grades are equal to 1. For simplicity, it is assumed here that the secondary MFs reach the value 1 at only one point. As explained in Sect. 6.8, this means that a T1 FS can be thought of as a special case of a GT2 FS; i.e., the case where only one value of the secondary MF has a secondary grade equal to 1 and all other secondary grades are equal to zero. Consequently, *if there are any set-theoretic laws that are not satisfied by T1 FSs it can safely be said that T2 FSs will not satisfy those laws either; however, the converse of this statement may not be true. If any condition is satisfied by T1 FSs, it may or may not be satisfied by GT2 FSs.*

This could be a very long appendix if it provided the details for all of the 20 laws (some have two or even four versions), four kinds of secondary MFs and two t-norms (a total of 160 calculations). Most of this is left as exercises for the reader. *Below some of the details are given but only for the product t-norm and normal and convex secondary MFs* (Table 7.2).

Consider three T2 FSs, $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ defined on the same universe of discourse, X, with secondary MFs as follows, for $x \in X$:

$$\mu_{\tilde{A}(x)}(u) = \int_{u \in [0,1]} f_x(u)/u \tag{7.116}$$

$$\mu_{\tilde{B}(x)}(u) = \int_{u \in [0,1]} g_x(u)/u \tag{7.117}$$

$$\mu_{\tilde{C}(x)}(u) = \int_{u \in [0,1]} h_x(u)/u \tag{7.118}$$

To begin, our attention is directed at the laws that are satisfied. As in Klir and Yuan (1995), fuzzy set inclusion is defined to mean:

$$\tilde{A} \subseteq \tilde{B} \Leftrightarrow \mu_{\tilde{A}(x)} \leq \mu_{\tilde{B}(x)} \quad x \in X \tag{7.119}$$

Since this does not make use of the t-norm or t-conorm, it is satisfied. The generalized versions of reflexive, anti-symmetric, and transitive laws, as stated in Tables 7.1 and 7.2, are also satisfied because they do not make use of the t-norm or t-conorm. Note that, in these laws, $\mu_{\tilde{A}(x)} = \mu_{\tilde{B}(x)} \Leftrightarrow f_x(u) = g_x(u)$, $u \in [0, 1]$.

Commutative laws are satisfied for maximum t-conorm and product t-norm (Mizumoto and Tanaka 1976), e.g., using (7.3):

$$\begin{aligned}
\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)} &= \left[\int_{u \in [0,1]} f_x(u)/u \right] \sqcup \left[\int_{v \in [0,1]} g_x(v)/v \right] \\
&= \int_{u \in [0,1]} \int_{v \in [0,1]} f_x(u) \times g_x(v)/(u \vee v) = \int_{u \in [0,1]} \int_{v \in [0,1]} g_x(v) \times f_x(u)/(v \vee u) \\
&= \left[\int_{v \in [0,1]} g_x(v)/v \right] \sqcup \left[\int_{u \in [0,1]} f_x(u)/u \right] = \mu_{\tilde{B}(x)} \sqcup \mu_{\tilde{A}(x)}
\end{aligned} \tag{7.120}$$

Associative laws are also satisfied for maximum t-conorm and product t-norm (Mizumoto and Tanaka 1976), e.g., using (7.7):

$$\begin{aligned}
\mu_{\tilde{A}(x)} \sqcap \left(\mu_{\tilde{B}(x)} \sqcap \mu_{\tilde{C}(x)} \right) &= \left[\int_{u \in [0,1]} f_x(u)/u \right] \sqcap \left[\int_{v \in [0,1]} \int_{w \in [0,1]} g_x(v) \times h_x(w)/(v \times w) \right] \\
&= \int_{u \in [0,1]} \int_{v \in [0,1]} \int_{w \in [0,1]} f_x(u) \times [g_x(v) \times h_x(w)]/u \times (v \times w) \\
&= \int_{u \in [0,1]} \int_{v \in [0,1]} \int_{w \in [0,1]} [f_x(u) \times g_x(v)] \times h_x(w)/(u \times v) \times w \\
&= \left[\int_{u \in [0,1]} \int_{v \in [0,1]} f_x(u) \times g_x(v)/(u \times v) \right] \sqcap \left[\int_{w \in [0,1]} h_x(w)/w \right] \\
&= \left(\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)} \right) \sqcap \mu_{\tilde{C}(x)}
\end{aligned} \tag{7.121}$$

The identity laws are all satisfied for maximum t-conorm and product t-norm if one uses normal membership grades, because:

$$\begin{aligned}\mu_{\tilde{A}(x)} \sqcup (1/0) &= \left[\int_{u\in[0,1]} f_x(u)/u\right] \sqcup 1/0 \\ &= \int_{u\in[0,1]} f_x(u) \times 1/(u \vee 0) = \int_{u\in[0,1]} f_x(u)/u = \mu_{\tilde{A}(x)}\end{aligned} \tag{7.122}$$

$$\begin{aligned}\mu_{\tilde{A}(x)} \sqcap (1/1) &= \left[\int_{u\in[0,1]} f_x(u)/u\right] \sqcap 1/1 \\ &= \int_{u\in[0,1]} f_x(u) \times 1/(u \times 1) = \int_{u\in[0,1]} f_x(u)/u = \mu_{\tilde{A}(x)}\end{aligned} \tag{7.123}$$

$$\begin{aligned}\mu_{\tilde{A}(x)} \sqcup (1/1) &= \left[\int_{u\in[0,1]} f_x(u)/u\right] \sqcup 1/1 = \int_{u\in[0,1]} f_x(u) \times 1/(u \vee 1) \\ &= \int_{u\in[0,1]} f_x(u)/1 = \left[\sup_u f_x(u)\right]/1 = 1/1\end{aligned} \tag{7.124}$$

$$\begin{aligned}\mu_{\tilde{A}(x)} \sqcap (1/0) &= \left[\int_{u\in[0,1]} f_x(u)/u\right] \sqcap 1/0 = \int_{u\in[0,1]} f_x(u) \times 1/(u \times 0) \\ &= \int_{u\in[0,1]} f_x(u)/0 = \left[\sup_u f_x(u)\right]/0 = 1/0\end{aligned} \tag{7.125}$$

The fact that the secondary MFs are normal only had to be used in (7.124) and (7.125). It was not used in the proofs of (7.122) and (7.123); hence, those identity laws hold even in the case of non-normal secondary MFs. Convexity of $\mu_{\tilde{A}(x)}$ played no role in these calculations.

Next, out attention is directed at the laws that are not satisfied. Examining (7.124) and (7.125), these laws fail when $\tilde{A}(x)$ is non-normal. The following counterexample demonstrates that the first parts of the idempotent and distributive laws, and the second part of the absorption law, which are satisfied in the type-1 case, are not satisfied for maximum t-conorm and product t-norm in the type-2 case.

Example 7.27 Consider the following three normal convex type-1 secondary MFs:

$$\mu_{\tilde{A}(x)}(u) = 0.5/0.1 + 1/0.7 \tag{7.126}$$

$$\mu_{\tilde{B}(x)}(u) = 0.6/0.3 + 1/0.7 \tag{7.127}$$

$$\mu_{\tilde{C}(x)}(u) = 0.4/0.2 + 1/0.8 \tag{7.128}$$

- *Failure of the first part of the idempotent law under product t-norm*:

$$\begin{aligned}\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{A}(x)} &= (0.5/0.1 + 1/0.7) \sqcup (0.5/0.1 + 1/0.7) \\ &= 0.25/0.1 + 0.5/0.7 + 0.5/0.7 + 1/0.7 \\ &= 0.25/0.1 + 1/0.7 \neq \mu_{\tilde{A}(x)}\end{aligned} \tag{7.129}$$

- *Failure of the first part of the distributive law under product t-norm* (details are left to the reader in Exercise 7.44):

$$\begin{aligned}\mu_{\tilde{A}(x)} \sqcap \left(\mu_{\tilde{B}(x)} \sqcup \mu_{\tilde{C}(x)}\right) = {} & 0.12/0.03 + 0.2/0.07 + 0.5/0.08 \\ & + 0.24/0.21 + 0.4/0.49 + 1/0.56\end{aligned} \tag{7.130}$$

$$\begin{aligned}\left(\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}\right) \sqcup \left(\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{C}(x)}\right) = {} & 0.06/0.03 + 0.1/0.07 + 0.25/0.08 \\ & + 0.2/0.14 + 0.3/0.21 + 0.5/0.49 + 1/0.56\end{aligned} \tag{7.131}$$

Comparing (7.130) and (7.131), one concludes that $\left(\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}\right) \sqcup \left(\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{C}(x)}\right) \neq \mu_{\tilde{A}(x)} \sqcap \left(\mu_{\tilde{B}(x)} \sqcup \mu_{\tilde{C}(x)}\right)$.

- *Failure of the second part of the absorption law under product t-norm* (details are left to the reader in Exercise 7.44):

$$\mu_{\tilde{A}(x)} \sqcup \left(\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}\right) = 0.25/0.1 + 0.3/0.21 + 0.5/0.49 + 1/0.7 \neq \mu_{\tilde{A}(x)} \tag{7.132}$$

Observe from Table 7.2 that both of the distributive and absorption laws do not hold. So, under maximum t-conorm and product t-norm, *meet is totally non-distributive and nonabsorptive for all GT2 FSs*.

Appendix 2: Proofs

2.1 Proof of Theorem 7.1

The union of $\tilde{A}$ and $\tilde{B}$ is another GT2 FS, just as the union of T1 FSs A and B is another T1 FS; hence, using the vertical-slice representation of a GT2 FS in (6.38) and (6.39) applied to $\tilde{A} \cup \tilde{B}$, it follows that:

$$\tilde{A} \cup \tilde{B} = \int_{x \in X} \mu_{(\tilde{A} \cup \tilde{B})_x}(u) \Big/ x \tag{7.133}$$

where

$$\mu_{(\tilde{A} \cup \tilde{B})_x}(u) = \int_{u \in [0,1]} h_x(u)/u \tag{7.134}$$

In (7.134), making use of (7.1) and (7.2),

$$\int_{u \in [0,1]} h_x(u)/u = \varphi\left(\mu_{\tilde{A}(x)}, \mu_{\tilde{B}(x)}\right) = \varphi\left(\int_{v \in [0,1]} f_x(v)/v, \int_{w \in [0,1]} g_x(w)/w\right) \tag{7.135}$$

Applying (2.71) to the right-hand side of (7.135), it follows that:

$$\varphi\left(\int_{v \in [0,1]} f_x(v)/v, \int_{w \in [0,1]} g_x(w)/w\right) = \int_{v \in [0,1]} \int_{w \in [0,1]} f_x(v) \bigstar g_x(w)/\varphi(v, w) \tag{7.136}$$

where φ is a t-conorm, chosen here as the maximum operation $\vee$, so that, from (7.134) to (7.136), it follows that:

$$\mu_{(\tilde{A} \cup \tilde{B})_x}(u) = \int_{u \in [0,1]} h_x(u)/u = \int_{v \in [0,1]} \int_{w \in [0,1]} f_x(v) \bigstar g_x(w) \Big/ (\underbrace{v \vee w}_{u}) \quad x \in X \tag{7.137}$$

In this equation $\iint$ indicates union over $[0, 1] \times [0, 1]$.

Applying (7.137) to (7.133), the result in (7.3) follows.

2.2 Proof of Theorem 7.3[19]

(a) Using (7.3) $F_1 \sqcup F_2$ can be expressed as:

$$F_1 \sqcup F_2 = \int_{v\in[0,1]} \int_{w\in[0,1]} [f_1(v)\bigstar f_2(w)/(\theta = v \vee w) \tag{7.138}$$

If $\theta \in F_1 \sqcup F_2$, the possible $\{v, w\}$ pairs that can give θ as the result of the maximum operation are $\{v, \theta\}$ where $v \in (-\infty, \theta]$ and $\{\theta, w\}$ where $w \in (-\infty, \theta]$. The process of finding the membership of θ in $F_1 \sqcup F_2$ can be broken into three steps: (1) find the t-norm (minimum or product) between the memberships of all the pairs $\{v, \theta\}$ such that $v \in (-\infty, \theta]$ and then find their supremum; (2) do the same with all the pairs $\{\theta, w\}$ such that $w \in (-\infty, \theta]$; and (3) find the maximum of the two suprema, i.e.

$$\mu_{F_1\sqcup F_2}(\theta) = \phi_1(\theta) \vee \phi_2(\theta) \tag{7.139}$$

where

$$\phi_1(\theta) = \sup_{v\in(-\infty,\theta]} \{f_1(v)\bigstar f_2(\theta)\} = f_2(\theta)\bigstar \sup_{v\in(-\infty,\theta]} f_1(v) \tag{7.140}$$

and

$$\phi_2(\theta) = \sup_{w\in(-\infty,\theta]} \{f_1(\theta)\bigstar f_2(w)\} = f_1(\theta)\bigstar \sup_{w\in(-\infty,\theta]} f_2(w) \tag{7.141}$$

To continue, θ is broken into the following three ranges: $\theta < \xi_1$, $\xi_1 \le \theta < \xi_2$ and $\theta \ge \xi_2$ (see Fig. 7.17a). From the normality of F_i $(i = 1, 2)$, $f_i(\xi_i) = 1$, and, from the convexity of F_i, f_i is monotonic nondecreasing in $(-\infty, \xi_i]$ and monotonic nonincreasing in $[\xi_i, \infty)$.

(i) $\theta < \xi_1$: See Fig. 7.17a, where θ is called θ_1. Because f_1 and f_2 both are monotonic nondecreasing in $(-\infty, \xi_1]$, $\sup_{v\in(-\infty,\theta]} f_1(v) = f_1(\theta)$ and $\sup_{w\in(-\infty,\theta]} f_2(w) = f_2(\theta)$; therefore, from (7.140) and (7.141), it follows that $\phi_1(\theta) = \phi_2(\theta) = f_1(\theta)\bigstar f_2(\theta)$, so that:

$$\mu_{F_1\sqcup F_2}(\theta) = f_1(\theta)\bigstar f_2(\theta), \theta < \xi_1 \tag{7.142}$$

---

[19]This very technical proof does not appear in Mendel (2001) because GT2 FSs were not popular in 2001; it is included here because GT2 FSs are now popular. It is adapted from Karnik and Mendel (2001, pp. 340–343).

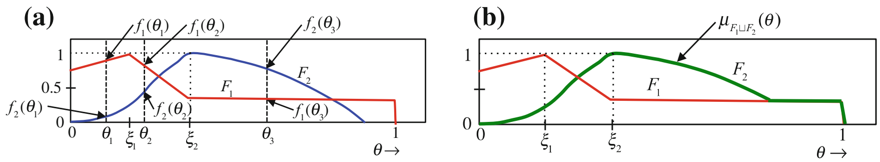


Fig. 7.17 **a** An example of F_1 and F_2 used in the proof of Part (**a**) of Theorem 7.3 [adapted from Fig. 6a in Karnik and Mendel (2001)]; and **b** $\mu_{F_1 \sqcup F_2}(\theta)$ using the minimum t-norm

(ii) $\xi_1 \leq \theta < \xi_2$: See Fig. 7.17a, where θ is called θ_2. Because $f_1(\xi_1) = 1$ and f_2 is monotonic nondecreasing in $(-\infty, \xi_2]$, $\sup_{v\in(-\infty,\theta]} f_1(v) = 1$ and $\sup_{w\in(-\infty,\theta]} f_2(w) = f_2(\theta)$. Using these facts in (7.139)–(7.141), it follows that[20]:

$$\mu_{F_1 \sqcup F_2}(\theta) = f_2(\theta) \vee [f_1(\theta) \wedge f_2(\theta)] = f_2(\theta), \xi_1 \leq \theta < \xi_2 \tag{7.143}$$

(iii) $\theta \geq \xi_2$: See Fig. 7.17a, where θ is called θ_3. For θ in this range, both f_1 and f_2 have already attained their maximum values; therefore $\sup_{v\in(-\infty,\theta]} f_1(v) = \sup_{w\in(-\infty,\theta]} f_2(w) = 1$. Consequently, from (7.139) to (7.141), it follows that:

$$\mu_{F_1 \sqcup F_2}(\theta) = f_1(\theta) \vee f_2(\theta), \theta \geq \xi_2 \tag{7.144}$$

Collecting (7.142)–(7.144) together, $\mu_{F_1 \sqcup F_2}(\theta)$ can be expressed, as:

$$\mu_{F_1 \sqcup F_2}(\theta) = \begin{cases} f_1(\theta) \bigstar f_2(\theta), & \theta < \xi_1 \\ f_2(\theta), & \xi_1 \leq \theta < \xi_2 \\ f_1(\theta) \vee f_2(\theta), & \theta \geq \xi_2 \end{cases} \tag{7.145}$$

which is (7.5).

Because both F_1 and F_2 are convex and normal, $F_1 \sqcup F_2$ is convex and normal, via Theorem 7.2. It is also true that $\mu_{F_1 \sqcup F_2}(\xi_2) = 1$ (see Fig. 7.17b).

(b) To prove (7.6), begin with $n = 3$, use the associative law for three T1 FSs (namely, $F_1 \sqcup F_2 \sqcup F_3 = (F_1 \sqcup F_2) \sqcup F_3$), the fact that $F_1 \sqcup F_2$ is convex and normal with $\mu_{F_1 \sqcup F_2}(\xi_2) = 1$, and the given facts that $f_3(\xi_3) = 1$ and $\xi_3 \geq \xi_2$, so that (7.145) can be directly applied to compute $\mu_{(F_1 \sqcup F_2) \sqcup F_3}(\theta)$, as:

[20]For the minimum t-norm, it is well known that $a \vee [b \wedge a] = a$, for real a and b. For the product t-norm, because $f_1(\theta) \leq 1$ when $\xi_1 \leq \theta < \xi_2$, it follows that $f_1(\theta) \times f_2(\theta) \leq f_2(\theta)$, so that $f_2(\theta) \vee [f_1(\theta) \times f_2(\theta)] = f_2(\theta)$.

$$\mu_{F_1 \sqcup F_2 \sqcup F_3}(\theta) = \begin{cases} \mu_{F_1 \sqcup F_2}(\theta) \bigstar f_3(\theta), & \theta < \xi_2 \\ f_3(\theta), & \xi_2 \le \theta < \xi_3 \\ \mu_{F_1 \sqcup F_2}(\theta) \vee f_3(\theta), & \theta \ge \xi_3 \end{cases} \tag{7.146}$$

Because $\xi_2 \le \xi_3$, (7.145) and (7.146) can be rewritten as:

$$\mu_{F_1 \sqcup F_2}(\theta) = \begin{cases} f_1(\theta) \bigstar f_2(\theta), & \theta < \xi_1 \\ f_2(\theta), & \xi_1 \le \theta < \xi_2 \\ f_1(\theta) \vee f_2(\theta), & \xi_2 \le \theta < \xi_3 \\ f_1(\theta) \vee f_2(\theta), & \theta \ge \xi_3 \end{cases} \tag{7.147}$$

$$\mu_{F_1 \sqcup F_2 \sqcup F_3}(\theta) = \begin{cases} \mu_{F_1 \sqcup F_2}(\theta) \bigstar f_3(\theta), & \theta < \xi_1 \\ \mu_{F_1 \sqcup F_2}(\theta) \bigstar f_3(\theta), & \xi_1 \le \theta < \xi_2 \\ f_3(\theta), & \xi_2 \le \theta < \xi_3 \\ \mu_{F_1 \sqcup F_2}(\theta) \vee f_3(\theta), & \theta \ge \xi_3 \end{cases} \tag{7.148}$$

Substituting for $\mu_{F_1 \sqcup F_2}(\theta)$ from (7.147) into (7.148), it follows that:

$$\mu_{F_1 \sqcup F_2 \sqcup F_3}(\theta) = \begin{cases} f_1(\theta) \bigstar f_2(\theta) \bigstar f_3(\theta), & \theta < \xi_1 \\ f_2(\theta) \bigstar f_3(\theta), & \xi_1 \le \theta < \xi_2 \\ f_3(\theta), & \xi_2 \le \theta < \xi_3 \\ f_1(\theta) \vee f_2(\theta) \vee f_3(\theta), & \theta \ge \xi_3 \end{cases} \tag{7.149}$$

(7.149) can be reexpressed as:

$$\mu_{F_1 \sqcup F_2 \sqcup F_3}(\theta) = \begin{cases} T_{i=1}^{3} f_i(\theta), & \theta < \xi_1 \\ T_{i=k+1}^{3} f_i(\theta), & \xi_k \le \theta < \xi_{k+1},\ 1 \le k \le 2 \\ \vee_{i=1}^{3} f_i(\theta), & \theta \ge \xi_3 \end{cases} \tag{7.150}$$

(7.150) is the same as (7.6) for $n = 3$.

It is straightforward to show that $F_1 \sqcup F_2 \sqcup F_3$ is also a convex and normal T1 FS with $\mu_{F_1 \sqcup F_2 \sqcup F_3}(\xi_3) = 1$; therefore, (7.145) can be applied again to obtain $\mu_{F_1 \sqcup F_2 \sqcup F_3 \sqcup F_4}$. Continuing in this fashion (7.6) is obtained for all values of n.

2.3 Proof of Theorem 7.4

The intersection of $\tilde{A}$ and $\tilde{B}$ is another GT2 FS, just as the intersection of T1 FSs A and B is another T1 FS (Karnik and Mendel 2001, p. 343); hence, again using the vertical-slice representation of a GT2 FS in (6.38) and (6.39) applied to $\tilde{A} \cap \tilde{B}$, it follows that

$$\tilde{A} \cap \tilde{B} = \int_{x \in X} \mu_{(\tilde{A} \cap \tilde{B})_x}(u) \Big/ x \tag{7.151}$$

where

$$\mu_{(\tilde{A} \cap \tilde{B})_x}(u) = \int_{u \in [0,1]} h_x(u)/u \tag{7.152}$$

Because the development of $\mu_{(\tilde{A} \cap \tilde{B})_x}$ is the same as the development of $\mu_{(\tilde{A} \cup \tilde{B})_x}$, except that in the present case φ is the minimum or product function (again, corresponding to the type-1 case), namely $\wedge$, the rest of this proof is left to the reader (Exercise 7.6).

2.4 Proof of Theorem 7.6

For the case $n = 2$, $F_1 \sqcap F_2$ can be expressed as (Karnik and Mendel 2001, p. 343):

$$F_1 \sqcap F_2 = \int_{v \in [0,1]} \int_{w \in [0,1]} [f_1(v) \wedge f_2(w)/(\theta = v \wedge w) \tag{7.153}$$

The only difference between this operation and the join operation in (7.138) is that $\theta \in F_1 \sqcap F_2$ is the result of the minimum operation on some pair $\{v, w\}$, such that $v \in F_1$ and $w \in F_2$; the possible such pairs being $\{v, \theta\}$ where $v \in [\theta, \infty)$ and $\{\theta, w\}$ where $w \in [\theta, \infty)$. The process of finding the membership of θ in $F_1 \sqcap F_2$ can be broken into three steps: (1) find the minima between the memberships of all the pairs $\{v, \theta\}$ such that $v \in [\theta, \infty)$ and then find their supremum; (2) do the same with all pairs $\{\theta, w\}$ such that $w \in [\theta, \infty)$; and, (3) find the maximum of the two suprema.

Because the rest of the proof is very similar to the proof of Theorem 7.3, it is left to the reader (Exercise 7.7).

2.5 Proof of Theorem 7.7

By a type-2 fuzzy singleton (see Definition 6.26), is meant a T2 FS whose MF $\mu_{\tilde{A}}(x, v)$ is

$$\mu_{\tilde{A}}(x, v) = \begin{cases} 1/1 & x = x' \\ 1/0 & \forall x \neq x' \end{cases} \tag{7.154}$$

Type-2 fuzzy set $\tilde{B}$ is described by (7.2). From (7.7), (7.154) and (7.2), and the last rows of Tables 7.1 and 7.2, it follows that, for minimum or product t-norms (for both of which $1 \bigstar g_{x'}(w) = g_{x'}(w)$ and $1 \wedge w = w$),

$$\begin{aligned} \mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)} &= \begin{cases} (1/1) \sqcap \tilde{B}(x') & x = x' \\ (1/0) \sqcap \tilde{B}(x') & x \neq x' \end{cases} = \begin{cases} \int_{w \in [0,1]} 1 \bigstar g_{x'}(w) / (1 \wedge w) & x = x' \\ 1/0 & x \neq x' \end{cases} \\ &= \begin{cases} \int_{w \in [0,1]} g_{x'}(w)/w & x = x' \\ 1/0 & x \neq x' \end{cases} = \begin{cases} \tilde{B}(x') & x = x' \\ 1/0 & x \neq x' \end{cases} \end{aligned} \tag{7.155}$$

2.6 Proof of Theorem 7.8

The complement of $\tilde{A}$ is another T2 FS just as the complement of T1 FS A is another T1 FS; hence, again from (6.38) and (6.39) applied to $\bar{\tilde{A}}$, it follows that

$$\bar{\tilde{A}} = \int_{x \in X} \mu_{(\bar{\tilde{A}})_x}(u) \Big/ x \tag{7.156}$$

where

$$\mu_{(\bar{\tilde{A}})_x}(u) = \int_{u \in [0,1]} h_x(u)/u \tag{7.157}$$

In (7.157),

$$\int_{u \in [0,1]} h_x(u)/u = \varphi(\mu_{\tilde{A}(x)}) = \varphi\left(\int_{u \in [0,1]} f_x(u)/u \right) \tag{7.158}$$

Applying the (2.71) Extension Principle, but for a single variable, to the right-hand side of (7.158), it follows that:

$$\varphi\left(\int_{u\in[0,1]} f_x(u)/u\right) = \int_{u\in[0,1]} f_x(u)/\varphi(u) \tag{7.159}$$

where $\varphi(u)$ is the complement operation $1-u$, so that, from (7.157) to (7.159), it follows that:

$$\mu_{\left(\bar{\tilde{A}}\right)_x}(u) = \int_{u\in[0,1]} h_x(u)/u = \int_{u\in[0,1]} f_x(u)/(1-u) \quad x\in X \tag{7.160}$$

Applying (7.160) to (7.156), the result in (7.18) follows.

2.7 Proof of Theorem 7.11

The proof uses mathematical induction. (a) Let F_1 and F_2 be two type-1 interval fuzzy numbers with domains $[l_1,r_1]$ and $[l_2,r_2]$, respectively. The join between F_1 and F_2, under either minimum or product t-norm, can be expressed [using (7.138)], as:

$$F_1 \sqcup F_2 = \int_{v\in[l_1,r_1]} \int_{w\in[l_2,r_2]} 1/(v\vee w) \tag{7.161}$$

To begin, $r_1 \vee r_2$ is shown to be the right-most point in $F_1 \sqcup F_2$. To see this, note that for any $v\in[l_1,r_1]$ and $w\in[l_2,r_2]$, since $w\le r_2$, then $v\vee w=\max(v,w) \le v\vee r_2$, and since $v\le r_1$, $v\vee w\le r_1\vee r_2$.

Next, $l_1\vee l_2$ is shown to be the left-most point in $F_1 \sqcup F_2$. To see this, note that for any $v\in[l_1,r_1]$ and $w\in[l_2,r_2]$, since $w\ge l_2$, then $v\vee w=\max(v,w)\ge v\vee l_2$, and since $v\ge l_1$, $v\vee w\ge l_1\vee l_2$.

In summary, each term in $F_1 \sqcup F_2$ is equal to $v\vee w$ for some $v\in[l_1,r_1]$ and $w\in[l_2,r_2]$, the smallest term being $l_1\vee l_2$ and the largest term being $r_1\vee r_2$. Because F_1 and F_2 have continuous domains, $F_1 \sqcup F_2$ also has a continuous domain; hence, $F_1 \sqcup F_2$ is a type-1 interval fuzzy number with domain $[l_1\vee l_2, r_1\vee r_2]$.

(b) Assume that when $n=k$ this theorem is valid, i.e.,

$$F_1 \sqcup F_2 \sqcup \cdots \sqcup F_k = \int_{q\in[(l_1\vee l_2\vee\cdots\vee l_k),(r_1\vee r_2\vee\cdots\vee r_k)]} 1/q; \tag{7.162}$$

then, when $n = k + 1$,

$$
\begin{aligned}
F_1 \sqcup F_2 \sqcup \cdots \sqcup F_k \sqcup F_{k+1} &= (F_1 \sqcup F_2 \sqcup \cdots \sqcup F_k) \sqcup F_{k+1} \\
&= \left[\int_{q \in [(l_1 \vee l_2 \vee \cdots \vee l_k),(r_1 \vee r_2 \vee \cdots \vee r_k)]} 1/q \right] \sqcup \left[\int_{w \in [l_{k+1}, r_{k+1}]} 1/w \right] \\
&= \int_{g \in [(l_1 \vee l_2 \vee \cdots \vee l_k) \vee l_{k+1},(r_1 \vee r_2 \vee \cdots \vee r_k) \vee r_{k+1}]} 1/g \\
&= \int_{g \in [(l_1 \vee l_2 \vee \cdots \vee l_k \vee l_{k+1}),(r_1 \vee r_2 \vee \cdots \vee r_k \vee r_{k+1})]} 1/g
\end{aligned}
\tag{7.163}
$$

Equation (7.163) has made use of (7.161) and is what is obtained by setting $n = k + 1$ in the statement of the theorem; hence, this demonstrates that Theorem 7.11 is valid for all n.

2.8 Proof of Theorem 7.12

(a) *minimum t-norm*—In this case, the proof of Theorem 7.12 is so similar to the proof of Theorem 7.11 (making use of mathematical induction) that it is left as an exercise for the reader (Exercise 7.19).

(b) *product t-norm*—Let F_1 and F_2 be two interval type-1 fuzzy sets with domains $[l_1, r_1]$ and $[l_2, r_2]$, respectively. The meet between F_1 and F_2, under product t-norm, can be expressed [using (7.153)] as

$$F_1 \sqcap F_2 = \int_{v \in [l_1, r_1]} \int_{w \in [l_2, r_2]} (1 \times 1)/(vw) \tag{7.164}$$

Observe from (7.164) that: (1) each term in $F_1 \sqcap F_2$ is equal to the product vw for some $v \in [l_1, r_1]$ and $w \in [l_2, r_2]$; (2) since l_1, l_2, r_1 and r_2 are greater than or equal to zero, the smallest vw-product is $l_1 l_2$ and the largest vw-product is $r_1 r_2$; and (3) since both F_1 and F_2 have continuous domains, $F_1 \sqcap F_2$ also has a continuous domain; hence, $F_1 \sqcap F_2$ is a type-1 interval fuzzy number with domain $[l_1 l_2, r_1 r_2]$.

The rest of the proof again uses mathematical induction, is very straightforward, and is left to the reader (Exercise 7.19).

2.9 Proof of Theorem 7.14

Beginning with (7.52) and (7.53), it follows that:

$$\tilde{A} \cup \tilde{B} = \left(\int_{x \in X} \left[\sup_{\alpha \in [0,1]} \alpha / [a_\alpha(x), b_\alpha(x)] \right] / x \right) \cup \left(\int_{x \in X} \left[\sup_{\alpha \in [0,1]} \alpha / [c_\alpha(x), d_\alpha(x)] \right] / x \right) \tag{7.165}$$

Equation (7.165) can be reorganized, as follows:

$$\tilde{A} \cup \tilde{B} = \int_{x \in X} \left[\sup_{\alpha \in [0,1]} \alpha / [a_\alpha(x), b_\alpha(x)] \cup \sup_{\alpha \in [0,1]} \alpha / [c_\alpha(x), d_\alpha(x)] \right] \Big/ x \tag{7.166}$$

$$\tilde{A} \cup \tilde{B} = \int_{x \in X} \sup_{\alpha \in [0,1]} [\alpha / \{[a_\alpha(x), b_\alpha(x)] \cup [c_\alpha(x), d_\alpha(x)]\}] \Big/ x \tag{7.167}$$

$$\tilde{A} \cup \tilde{B} = \int_{x \in X} \sup_{\alpha \in [0,1]} [\alpha / \{[a_\alpha(x) \vee c_\alpha(x), b_\alpha(x) \vee d_\alpha(x)]\}] \Big/ x \tag{7.168}$$

which is the middle portion of (7.54). Note that, in going from (7.166) to (7.167) use has been made of (2.81).

Equation (7.167) can also be reexpressed as:

$$\tilde{A} \cup \tilde{B} = \sup_{\alpha \in [0,1]} \alpha \Big/ \left[\int_{x \in X} [a_\alpha(x), b_\alpha(x)] / x \right] \cup \left[\int_{x \in X} [c_\alpha(x), d_\alpha(x)] / x \right] \tag{7.169}$$

Applying (7.55) and (7.56) to (7.169), one obtains

$$\tilde{A} \cup \tilde{B} = \sup_{\alpha \in [0,1]} \alpha / (\tilde{A}_\alpha \cup \tilde{B}_\alpha) = \bigcup_{\alpha \in [0,1]} \alpha / (\tilde{A}_\alpha \cup \tilde{B}_\alpha) \tag{7.170}$$

which is the right-most side of (7.54).

2.10 Proof of Theorem 7.17

In this proof[21] Karnik and Mendel (2001, p. 338), analogous to the proof of Theorem 2.1, the following method is used: Let **A** be the statement "$\mu_{\tilde{R}\circ\tilde{S}}(u,w) \neq 1/0$," and **B** be the statement "there exists at least one $v \in V$ such that $\mu_{\tilde{R}}(u,v) \neq 1/0$ *and* $\mu_{\tilde{S}}(v,w) \neq 1/0$." It is proved that "**A** if and only if **B**" by first proving that $\bar{\mathbf{B}} \Rightarrow \bar{\mathbf{A}}$ (equivalent to proving that $\mathbf{A} \Rightarrow \mathbf{B}$, i.e., necessity of **B**) and then proving that $\bar{\mathbf{A}} \Rightarrow \bar{\mathbf{B}}$ (equivalent to proving that $\mathbf{B} \Rightarrow \mathbf{A}$, i.e., sufficiency of **B**).

Necessity—If there exists no $v \in V$ such that $\mu_{\tilde{R}}(u,v) \neq 1/0$ *and* $\mu_{\tilde{S}}(v,w) \neq 1/0$, then this means that for every $v \in V$, either $\mu_{\tilde{R}}(u,v) = 1/0$ or $\mu_{\tilde{S}}(v,w) = 1/0$ (or both are $1/0$). So, assuming that $\mu_{\tilde{R}}(u,v)$ and $\mu_{\tilde{S}}(v,w)$ are normal, then from the identity law $\mu_{\tilde{A}} \sqcap 1/0 = 1/0$ (see the last row in Tables 7.1 and 7.2 in Appendix 1, which requires $\mu_{\tilde{A}}$ to be normal), $\mu_{\tilde{R}}(u,v) \sqcap 1/0 = 1/0$ or $1/0 \sqcap \mu_{\tilde{S}}(v,w) = 1/0$, so that $\mu_{\tilde{R}}(u,v) \sqcap \mu_{\tilde{S}}(v,w) = 1/0$ for every $v \in V$, and therefore, in (7.87) $\sqcup_{v\in V}[1/0] = 1/0$ (see the first row of the Identity Law in Tables 7.1 and 7.2 in Appendix 1, in which $\tilde{A}(x) = 1/0$).

Sufficiency—If the extended sup-star composition is zero, then, from (7.87):

$$\sqcup_{v\in V}\left[\mu_{\tilde{R}}(u,v) \sqcap \mu_{\tilde{S}}(v,w)\right] = 1/0 \tag{7.171}$$

For each value of v, the term in the bracket is a T1 FS. Suppose, for example, that there are two such terms, P and Q, so that $\mu_P \sqcup \mu_Q = 1/0$. For two normal membership grades, expressed as $\mu_P = \int_{u\in U} f(u)/u$ and $\mu_Q = \int_{v\in V} g(v)/v$, it follows from (7.3) that

$$\mu_P \sqcup \mu_Q = 1/0 \Leftrightarrow \int\limits_{u\in U}\ \int\limits_{v\in V} f(u)\bigstar g(v)/(u \vee v) = 1/0 \tag{7.172}$$

For $u \vee v = 0$, it must be true that $u = v = 0$, so that for the right-hand side of (7.172) to hold, it must also be true that $\mu_P = \mu_Q = 1/0$. This means that each of the two terms in the bracket of (7.87) equals $1/0$.

The extension of this analysis to more than two terms is easy. One concludes, therefore, that $\mu_{\tilde{R}}(u,v) \sqcap \mu_{\tilde{S}}(v,w) = 1/0$ for every $v \in V$, which means that for every $v \in V$, either $\mu_{\tilde{R}}(u,v)$ or $\mu_{\tilde{S}}(v,w)$ (or both) is $1/0$. Consequently, there is no $v \in V$ such that $\mu_{\tilde{R}}(u,v) \neq 1/0$ and $\mu_{\tilde{S}}(v,w) \neq 1/0$.

[21] John et al. (2006) present an alternate proof by first defining *embedded type-2 fuzzy relations*, $\tilde{R}_e^k$, then using a wavy-slice representation of them, $\tilde{R} = \sum_{k=1}^{N} \tilde{R}_e^k$, in an *embedded sup-star composition*, $\mu_{[\tilde{R}\circ\tilde{S}]_e}(u,w) = \sqcup_{v\in V}\left[\mu_{\tilde{R}_e}(u,v) \sqcap \mu_{\tilde{S}_e}(v,w)\right]$, $u \in U$, $v \in V$.

Exercises

7.1 Repeat Example 7.1 using the product t-norm and maximum t-conorm.

7.2 As in Example 7.2, show that the extended operations for meet and negation reduce to the original ones when one deals with T1 FSs.

7.3 (a) In Theorem 7.2 prove that $\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}$ is normal. (b) In Theorem 7.5 prove that $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)}$ is normal.

7.4 In Example 7.3, develop the counterpart to the results shown in Fig. 7.1b, using the minimum t-norm.

7.5 Using F_1 and F_2 in Fig. 7.3, explain why Corollary 7.1 is not true for the product t-norm.

7.6 Complete the proof of Theorem 7.4.

7.7 Complete the proof of Theorem 7.6.

7.8 Try proving part (a) of Theorem 7.6 when $\star =$ product. Examine the proof, as was done in the proof of Theorem 7.3, for (see Fig. 7.17a): (i) $\theta < \xi_1$, (ii) $\xi_1 \leq \theta < \xi_2$ and (iii) $\theta \geq \xi_2$. For which of these steps does the proof break down, and why?

7.9 Using F_1 and F_2 in Fig. 7.3, explain why Corollary 7.2 is not true for the product t-norm.

7.10 Compute the join and meet of $\tilde{A}(x)$ and $\tilde{B}(x)$, when (Greenfield and John 2007): $\mu_{\tilde{A}(x)} = 0.3/0 + 0.8/0.25 + 1/0.5 + 0.5/0.75 + 0/1$ and $\mu_{\tilde{B}(x)} = 1/0 + 0.8/0.25 + 0.3/0.5 + 0.2/0.75 + 0.1/1$.

7.11 Suppose, for a particular element x, that $\mu_{\tilde{A}(x)} = 0.6/0.2 + 0.8/0.3 + 0.2/0.4$ and $\mu_{\tilde{B}(x)} = 0.7/0.2 + 0.3/0.5 + 0.1/0.8$. Compute $\mu_{(\tilde{A} \cup \tilde{B})_x}(u)$, $\mu_{(\tilde{A} \cap \tilde{B})_x}(u)$, $\mu_{(\bar{\tilde{A}})_x}(u)$, and $\mu_{(\bar{\tilde{B}})_x}(u)$.

7.12 In Example 7.14, verify (7.24) and (7.25).

7.13 This exercise focuses on two secondary MFs, one a triangle and the other a type-1 interval fuzzy number. Compute and sketch the join and meet between these secondary MFs, for the three situations that are shown in Fig. 7.18, using Theorems 7.3 and 7.6.

7.14
- (a) (Mendel 2011) For each of the nine situations shown in Table 7.3, show that the join is given by the heavy red curve.
- (b) Observe in Table 7.3 that each figure is labeled S1, or S2, … , or S5. What are five conclusions that can be drawn from the join results that are in this table?

7.15
- (a) (Mendel 2011) For each of the nine situations shown in Table 7.4, show that the meet is given by the heavy red curve.
- (b) Observe in Table 7.4 that each figure is labeled S1, or S2, …, or S5. What are five conclusions that can be drawn from the meet results that are in this table?

7.16 This exercise examines the meet between a *T1 FS*, A, and a *GT2 FS*, $\tilde{B}$, under minimum and product t-norms. To do this, A is represented as a GT2 FS (see

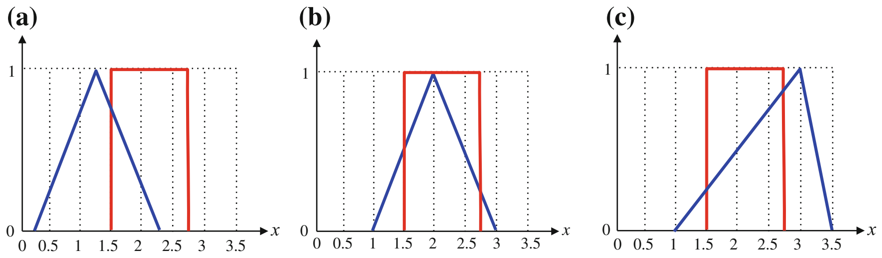


Fig. 7.18 Secondary MFs for Exercise 7.13

Table 7.3 Join examples for Exercise 7.14 (Mendel 2011: © 2011 IEEE)

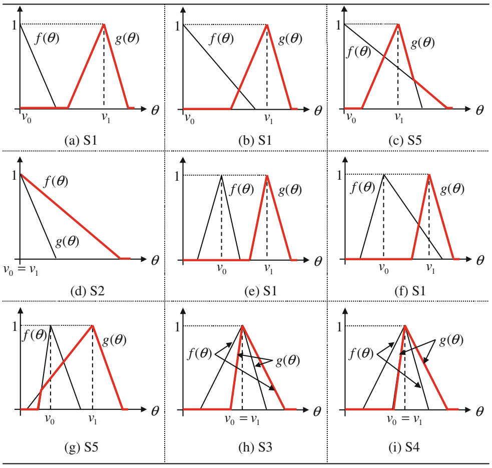
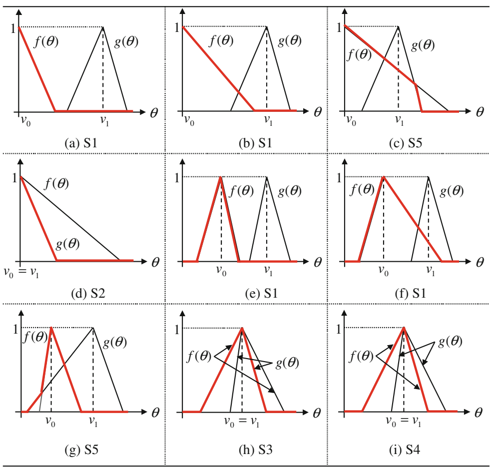


Table 7.4 Meet examples for Exercise 7.15 (Mendel 2011: © 2011 IEEE)



Sect. 6.8), as ($x \in X$): $\tilde{A} = 1/A \rightarrow \mu_{\tilde{A}} = 1/\mu_A$, and $\tilde{B}$ is described by its MF $\mu_{\tilde{B}}(x,u)$, where

$$\mu_{\tilde{B}}(x,u) = \int_{x \in X} \mu_{\tilde{B}(x)}(u) \Big/ x = \int_{x \in X} \left[\int_{u \in K \equiv [l,r] \subset [0,1]} g_x(u)/u \right] \Big/ x$$

(a) For *product t-norm*, show that $\mu_A(x) \sqcap \mu_{\tilde{B}(x)} = \int_{u \in K} g_x(u)/[\mu_A(x) \cdot u]$

(b) For *minimum t-norm*, the evaluation of $\mu_A(x) \sqcap \mu_{\tilde{B}(x)}$ requires considering the three cases that are depicted in Fig. 7.19. For case 1, $\mu_A(x) < u$ when $u \in K$; for case 2, $\mu_A(x) \in K$ when $u \in K$; and for case 3, $\mu_A(x) > u$ when $u \in K$. Show that:

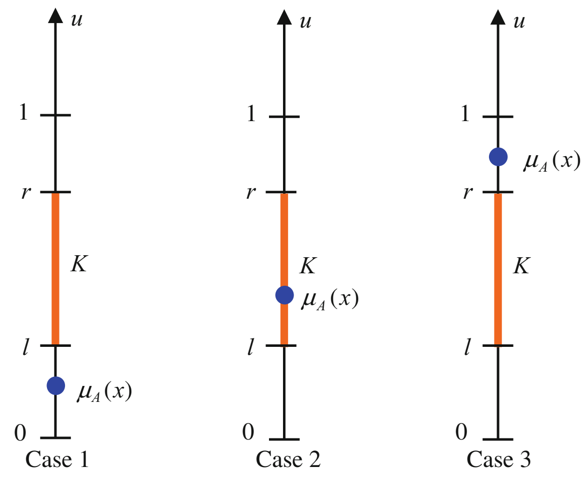
$$\mu_A(x) \sqcap \mu_{\tilde{B}(x)} = \begin{cases} 1/\mu_A(x) & \text{if } \mu_A(x) < u \quad \text{and } u \in K \\ \int_{d \in D} g_x(d)/d & \text{if } \mu_A(x) \in K \quad \text{and } u \in K \text{, and } D \equiv [l, \mu_A(x)] \\ \mu_{\tilde{B}(x)} & \text{if } \mu_A(x) > u \quad \text{and } u \in K \end{cases}$$

7.17 Figure 7.20 depicts a type-1 interval fuzzy number $g(v)$ and a triangular type-1 set $f(v)$. Note that L, R, A, T, and I stand for left, right, apex, triangle, and interval, respectively. Using (7.12) and (7.16) for the meet using the product t-norm, prove that $\mu_{F \sqcap G}(\theta)$ is given by the trapezoidal function that is depicted in Fig. 7.21.

7.18 Apply (7.6) to the case when the secondary MFs are type-1 interval fuzzy numbers, and check the results obtained with those given in Theorem 7.11.

7.19 (a) Prove part (a) in the proof of Theorem 7.12. (b) Complete the proof of part (b) in the proof of Theorem 7.12.

Fig. 7.19 Three cases for Exercise 7.16



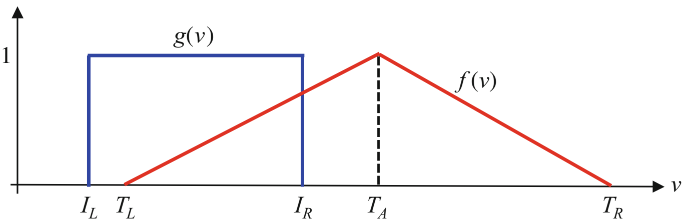


Fig. 7.20 Type-1 interval fuzzy number and triangular type-1 fuzzy set for Exercise 7.17

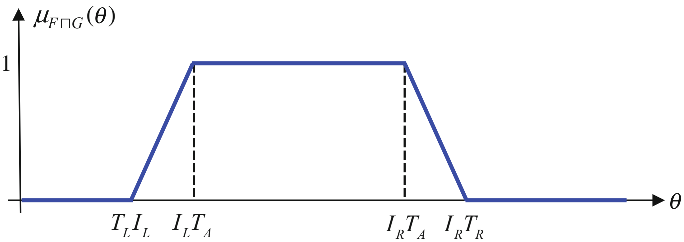


Fig. 7.21 Solution to Exercise 7.17

7.20 Apply (7.10) to the case when the secondary MFs are type-1 interval fuzzy numbers, and check the results obtained with those given in Theorem 7.12.

7.21 Let F_1 and F_2 be two type-1 interval fuzzy numbers, having domains $[a, b]$ and $[c, d]$, respectively. Assume that $[a, b]$ and $[c, d]$ do not overlap, i.e., $d < a$ or $c > b$. Compute $F_1 \sqcup F_2$ and $F_1 \sqcap F_2$ for both situations using the minimum t-norm. Draw some conclusions about the join and meet of two type-1 interval fuzzy numbers, whose domains do not overlap. Note that there do not appear to be comparable results for product t-norm.

7.22 Given the three type-1 interval fuzzy numbers, F_1, F_2 and F_3 having the domains $[0.1, 0.3]$, $[0.15, 0.25]$ and $[0.2, 0.4]$, respectively. Compute the following: (a) $F_1 \sqcup F_2 \sqcup F_3$ and (b) $F_1 \sqcap F_2 \sqcap F_3$. Do (b) for both minimum and product t-norms.

7.23 Formulas for the join and meet of IT2 FSs assumed in their derivations that supports of secondary MFs are connected (see Definitions 6.4 and 6.6). What happens in (7.161), for the join, when supports are disconnected? To answer this question assume that $\mu_{\tilde{A}(x')}$ and $\mu_{\tilde{B}(x')}$ have supports that are depicted in Fig. 7.22. Only consider the two cases when $f < a$ or $e > d$. Draw conclusions.

7.24 Redo Examples 7.1, 7.7 and 7.12 using horizontal slices, as explained in Sects. 7.4.1, 7.4.2 and 7.4.3, respectively.

7.25 This exercise is a generalization of Exercise 2.33 to IT2 FSs whose lower and upper MFs are trapezoids that may or may not be normal (Lee and Chen 2008; Wei and Chen 2009). Its results are finding applications because of their closed-form nature. Let (see Fig. 7.23)

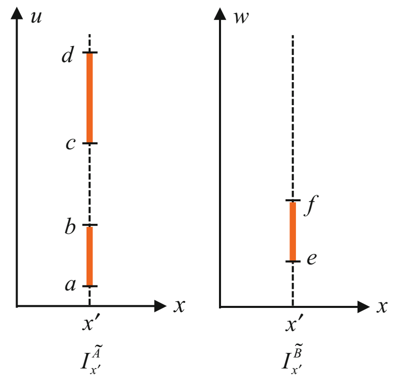


Fig. 7.22 Supports of secondary MFs for Exercise 7.23

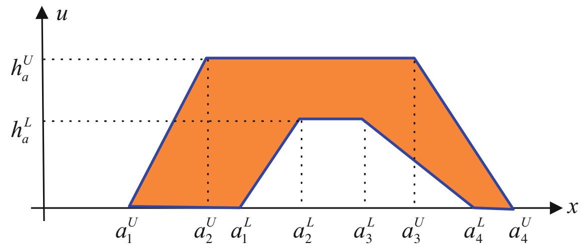


Fig. 7.23 Trapezoidal lower and upper MFs for Exercise 7.25

$$\tilde{A}_1 = (A_1^U, A_1^L) = \left(\left[a_{11}^U, a_{12}^U, a_{13}^U, a_{14}^U; h_{a_1}^U\right], \left[a_{11}^L, a_{12}^L, a_{13}^L, a_{14}^L; h_{a_1}^L\right]\right)$$

$$\tilde{A}_2 = (a_2^U, a_2^L) = \left(\left[a_{21}^U, a_{22}^U, a_{23}^U, a_{24}^U; h_{a_2}^U\right], \left[a_{21}^L, a_{22}^L, a_{23}^L, a_{24}^L; h_{a_2}^L\right]\right)$$

Prove that (see the end of Exercise 2.33 for explanation of $\approx$, which here applies to both the LMF and UMF):

(a) $$\tilde{A}_1 + \tilde{A}_2 \approx \left(\left[a_{11}^U + a_{21}^U, a_{12}^U + a_{22}^U, a_{13}^U + a_{23}^U, a_{14}^U + a_{24}^U; \min(h_{a_1}^U, h_{a_2}^U)\right],\right.$$
$$\left.\left[a_{11}^L + a_{21}^L, a_{12}^L + a_{22}^L, a_{13}^L + a_{23}^L, a_{14}^L + a_{24}^L; \min(h_{a_1}^L, h_{a_2}^L)\right]\right)$$

(b) $$\tilde{A}_1 - \tilde{A}_2 \approx \left(\left[a_{11}^U - a_{24}^U, a_{12}^U - a_{23}^U, a_{13}^U - a_{22}^U, a_{14}^U\right.\right.$$
$$\left. - a_{21}^U; \min\left(h_{a_1}^U, h_{a_2}^U\right)\right], \left[a_{11}^L - a_{24}^L, a_{12}^L - a_{23}^L, a_{13}^L\right.$$
$$\left.\left. - a_{22}^L, a_{14}^L - a_{21}^L; \min\left(h_{a_1}^L, h_{a_2}^L\right)\right]\right)$$

(c) $\tilde{A}_1 \cdot \tilde{A}_2 \approx \left(\left[a_{11}^U a_{21}^U, a_{12}^U a_{22}^U, a_{13}^U a_{23}^U, a_{14}^U a_{24}^U; \min(\mathrm{h}_{\mathrm{a}_1}^{\mathrm{U}}, \mathrm{h}_{\mathrm{a}_2}^{\mathrm{U}})\right],\right.$
$\left.\left[a_{11}^L a_{21}^L, a_{12}^L a_{22}^L, a_{13}^L a_{23}^L, a_{14}^L a_{24}^L; \min(\mathrm{h}_{\mathrm{a}_1}^{\mathrm{L}}, \mathrm{h}_{\mathrm{a}_2}^{\mathrm{L}})\right]\right)$

(d) $\tilde{A}_1^{-1} \approx \left(\left[1/a_{14}^U, 1/a_{13}^U, 1/a_{12}^U, 1/a_{11}^U; h_{a_1}^U\right], \left[1/a_{14}^L, 1/a_{13}^L, 1/a_{12}^L, 1/a_{11}^L; h_{a_1}^L\right]\right)$

(e) $\tilde{A}_1/\tilde{A}_2 \approx \left(\left[a_{11}^U/a_{24}^U, a_{12}^U/a_{23}^U, a_{13}^U/a_{22}^U, a_{14}^U/a_{21}^U; \min\left(h_{a_1}^U, h_{a_2}^U\right)\right],\right.$
$\left.\left[a_{11}^L/a_{24}^L, a_{12}^L/a_{23}^L, a_{13}^L/a_{22}^L, a_{14}^L/a_{21}^L; \min\left(h_{a_1}^L, h_{a_2}^L\right)\right]\right)$

(f) What do the formulas in (a)–(e) reduce to when $\tilde{A}_1$ and $\tilde{A}_2$ are perfectly normal interval type-2 fuzzy numbers?

7.26 Prove Theorem 7.15.
7.27 Prove Theorem 7.16.
7.28 Develop a procedure for computing $\neg\mu_{\tilde{A}(x)}$ using (7.70).
7.29 Complete the calculations to obtain the results in (7.85) and (7.86).
7.30 Explain how to carry out the computations in the extended sup-star composition (7.87) when U, W, and V are continuous universes of discourse.
7.31 Complete the calculations to obtain the results in (7.91).
7.32 Redo the calculations in Example 7.24 using product t-norm.
7.33 Carefully show how (7.92) follows from (7.87), when the first relation in the latter equation is just a GT2 FS.
7.34 Complete the calculations to obtain the results in (7.95).
7.35 Suppose $U = \{2, 12\}$, $V = \{1, 7, 13\}$ and

$$\mu_{\tilde{R}}(u) \equiv \begin{array}{cc} u_1 & u_2 \\ (1/0.7 + 1/0.9 & 1/0.1 + 1/0.4) \end{array}$$

$$\mu_{\tilde{S}}(u,v) \equiv \begin{array}{c} \\ u_1 \\ u_2 \end{array} \begin{array}{ccc} v_1 & v_2 & v_2 \\ \left(\begin{array}{c} 1/0.8 + 1/0.9 + 1/1 \\ 1/0.1 + 1/0.3 \end{array}\right. & \begin{array}{c} 1/0.3 + 1/0.4 + 1/0.5 \\ 1/0.2 + 1/0.5 + 1/0.8 \end{array} & \left.\begin{array}{c} 1/0 + 1/0.1 \\ 1/0.6 + 1/0.7 + 1/0.9 \end{array}\right) \end{array}.$$

Compute $\mu_{\tilde{R}\circ\tilde{S}}$ by using (7.92) and the join and meet formulas in (7.3) and (7.7).
7.36 Perform the detailed calculations to obtain $\mu_{(\tilde{F}\times\tilde{G})(x_1,x_2)}$ in (7.112).
7.37 Show that for GT2 FSs, whose secondary MFs are normal and convex, all the set-theoretic laws are satisfied under maximum t-conorm and minimum t-norm.
7.38 Explain why all of the "NO" elements in Table 7.1 are correct.
7.39 Show that for IT2 FSs the entries in the "Normal/Convex" column of Table 7.2 become the same as the entries in the "Product t-norm" column of Table 2-8.

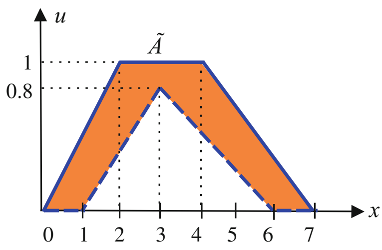


Fig. 7.24 IT2 FS for Exercise 7.45

7.40 Repeat Example 7.27 for the following normal and non-convex secondary MFs: $\mu_{\tilde{A}(x)}(u) = 0.5/0.1 + 0.2/0.3 + 1/0.7$, $\mu_{\tilde{B}(x)}(u) = 0.6/0.3 + 0.3/0.5 + 1/0.7$, $\mu_{\tilde{C}(x)}(u) = 0.4/0.2 + 0.2/0.6 + 1/0.8$.

7.41 Show that, under product t-norm and maximum t-conorm, $\mu_{\tilde{A}(x)} \sqcap \mu_{\tilde{B}(x)} = \mu_{\tilde{B}(x)} \sqcap \mu_{\tilde{A}(x)}$.

7.42 Show that, under product t-norm and maximum t-conorm, $\mu_{\tilde{A}(x)} \sqcup (\mu_{\tilde{B}(x)} \sqcup \mu_{\tilde{C}(x)}) = (\mu_{\tilde{A}(x)} \sqcup \mu_{\tilde{B}(x)}) \sqcup \mu_{\tilde{C}(x)}$.

7.43 Show that for GT2 FSs the involution law is satisfied under maximum t-conorm and product t-norm.

7.44 Complete all the details for Example 7.27.

7.45 This exercise is a generalization of the T1 FS cardinality Exercise 2.43 to IT2 FSs and GT2 FSs. It is based on results that are in Wu and Mendel (2007), Mendel (2009) and Zhai and Mendel (2011). The reader must read Exercise 2.43 before proceeding. Let $P_{\tilde{A}}$ denote the cardinality of IT2 FS $\tilde{A}$.

(a) Use the Wavy-Slice Representation Theorem 6.3 to prove that $P_{\tilde{A}} = [\mathrm{card}(\underline{\mu}_{\tilde{A}}(x)), \mathrm{card}(\bar{\mu}_{\tilde{A}}(x))]$.

(b) Using the T1 FS cardinality $p(A)$ that is given in Exercise 2.43, write formulas for $P_{\tilde{A}}$ for continuous and discrete universes of discourse.

(c) Compute $P_{\tilde{A}}$ for the IT2 FS that is depicted in Fig. 7.24, when $N = 8$.

(d) Associated with $P_{\tilde{A}}$ is the average cardinality of $\tilde{A}$, $AC(\tilde{A})$, which is a number, where $AC(\tilde{A}) = \frac{1}{2}[\mathrm{card}(\underline{\mu}_{\tilde{A}}(x)) + \mathrm{card}(\bar{\mu}_{\tilde{A}}(x))]$. Compute $AC(\tilde{A})$ for $\tilde{A}$ in Fig. 7.24.

(e) Explain how the cardinality of a GT2 FS can be defined and computed. Would the result be a number, T1 FS, IT2 FS or a GT2 FS? If it is not a number, then propose a way to obtain $AC(\tilde{A})$ for a GT2 FS. (Hint: See Sect. 7.12.)

7.46 This exercise is a generalization of the T1 FS *similarity* Exercise 2.44 to IT2 FSs and GT2 FSs. It is based on results that are in Wu and Mendel (2009) and Mendel and Wu (2010, Chap. 4). The reader must read Exercise 2.44 before proceeding. Just as there are many definitions for the similarity of T1 FSs, there are many (although not nearly so many as for T1 FSs) definitions for the similarity of IT2 FSs [see the two just mentioned references for discussions about this, as well as Livi et al. (2014)]. One could use the Wavy-Slice Representation Theorem 6.3, as was done in Exercise 7.45, to obtain a formula for the similarity of an IT2 FS; however, doing this would lead to a type-1 interval fuzzy number for such a similarity. This exercise prefers to examine a crisp number for the similarity of IT2 FSs.[22]

Let $sm(\tilde{A}, \tilde{B})$ denote the similarity between the two IT2 FSs $\tilde{A}$ and $\tilde{B}$. A crisp numerical Jaccard similarity measure between $\tilde{A}$ and $\tilde{B}$ is: $sm_J(\tilde{A}, \tilde{B}) = AC(\tilde{A} \cap \tilde{B}) / AC(\tilde{A} \cup \tilde{B})$, where AC is the average cardinality that is defined in Exercise 7.45.[23]

(a) Show that, for discrete universes of discourse:

$$sm_J(\tilde{A}, \tilde{B}) = \frac{\sum_{i=1}^{N} \min\left(\bar{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{B}}(x_i)\right) + \sum_{i=1}^{N} \min\left(\underline{\mu}_{\tilde{A}}(x_i), \underline{\mu}_{\tilde{B}}(x_i)\right)}{\sum_{i=1}^{N} \max\left(\bar{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{B}}(x_i)\right) + \sum_{i=1}^{N} \max\left(\underline{\mu}_{\tilde{A}}(x_i), \underline{\mu}_{\tilde{B}}(x_i)\right)}$$

(b) What is the comparable formula for continuous universes of discourse?
(c) Compute $sm_J(\tilde{A}, \tilde{B})$ for $\tilde{A}$ and $\tilde{B}$ that are depicted in Fig. 7.25.
(d) Explain how an analogous Jaccard similarity measure of a GT2 FS can be defined and computed. Will it be a single number or a type-1 fuzzy set? See Hao and Mendel (2014) for numerical examples and more discussions about this (Hint: See Sect. 7.12).

7.47 This exercise is a generalization of the T1 FS *subsethood* Exercise 2.45 to IT2 FSs and GT2 FSs. It is based on results that are in Mendel and Wu (2010, Chap. 4). The reader must read Exercise 2.45 before proceeding. Just as there are different definitions for the subsethood of T1 FSs, there are different (although not nearly so many as for T1 FSs) definitions for the subsethood of IT2 FSs (see Mendel and Wu (2010, Chap. 4) for discussions about this). One could use the Wavy-Slice Representation Theorem 6.3, as

[22]Please note that the use of a crisp number for the similarity of IT2 FSs is not being absolutely advocated for. Arguments can be given for using an interval similarity measure just as well as or for using a crisp number for similarity. The application may dictate which kind of measure is preferable. Of greater importance is that a similarity measure should satisfy some desirable properties, otherwise any kind of a measure between two IT2 FSs could be claimed to be a similarity measure. Four desirable properties for an IT2 FS similarity measure $sm(\tilde{A}, \tilde{B})$ are (Wu and Mendel 2009, Mendel and Wu 2010, Chap. 4): (1) *Reflexivity*: $sm(\tilde{A}, \tilde{B}) = 1 \Leftrightarrow \tilde{A} = \tilde{B}$; (2) *Symmetry*: $sm(\tilde{A}, \tilde{B}) = sm(\tilde{B}, \tilde{A})$; (3) *Transitivity*: If $\tilde{C} \leq \tilde{A} \leq \tilde{B}$ (Note: $\tilde{A} \leq \tilde{B}$ if $\bar{\mu}_{\tilde{A}}(x) \leq \bar{\mu}_{\tilde{B}}(x)$ and $\underline{\mu}_{\tilde{A}}(x) \leq \underline{\mu}_{\tilde{B}}(x)$ for $x \in X$) then $sm(\tilde{C}, \tilde{A}) \geq sm(\tilde{C}, \tilde{B})$; and (4) *Overlapping*: If $\tilde{A} \cap \tilde{B} \neq \varnothing$, then $sm(\tilde{A}, \tilde{B}) > 0$; otherwise, $sm(\tilde{A}, \tilde{B}) = 0$.

[23]A proof that this similarity measure satisfies all four of the desirable properties for an IT2 FS similarity measure, that are explained in Footnote 22, can be found in Wu and Mendel (2009) and Mendel and Wu (2010, Appendix 4B.1).

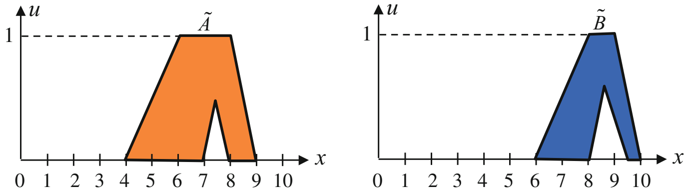


Fig. 7.25 Two FOUs for Exercises 7.46 and 7.47

was done in Exercise 7.45, to obtain a formula for the subsethood of an IT2 FS; however, doing this would lead to a type-1 interval fuzzy number for such a subsethood. This exercise prefers to examine a crisp number for the subsethood of IT2 FSs.[24]

Let $sm(\tilde{A},\tilde{B})$ denote the subsethood between the two IT2 FSs $\tilde{A}$ and $\tilde{B}$. The Vlachos and Sergiadis subsethood measure between $\tilde{A}$ and $\tilde{B}$ is $ss_{VS}(\tilde{A},\tilde{B}) = AC(\tilde{A}\cap\tilde{B})/AC(\tilde{A})$, where AC is the average cardinality that is defined in Exercise 7.45.[25]

(a) Show that, for discrete universes of discourse:

$$ss_{VS}(\tilde{A},\tilde{B}) = \frac{\sum_{i=1}^{N}\min\left(\underline{\mu}_{\tilde{A}}(x_i),\underline{\mu}_{\tilde{B}}(x_i)\right) + \sum_{i=1}^{N}\min\left(\bar{\mu}_{\tilde{A}}(x_i),\bar{\mu}_{\tilde{B}}(x_i)\right)}{\sum_{i=1}^{N}\underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=1}^{N}\bar{\mu}_{\tilde{A}}(x_i)}$$

(b) What is the comparable formula for continuous universes of discourse?

(c) Compute $ss_{VS}(\tilde{A},\tilde{B})$ for $\tilde{A}$ and $\tilde{B}$ that are depicted in Fig. 7.25.

(d) Explain how an analogous Vlachos and Sergiadis subsethood measure of a GT2 FS can be defined and computed. Will it be a single number or a T1 FS? (Hint: see Sect. 7.12.)

---

[24]Please note that the use of a crisp number for the subsethood of IT2 FSs is not being absolutely advocated for. Arguments can be given for using an interval subsethood measure [e.g., Cornelis and Kerre (2004), Nguyen and Kreinovich (2008), Rickard et al. (2008)] just as well as or for using a crisp number for subsethood. The application may dictate which kind of measure is preferable. Of greater importance is that a subsethood measure should satisfy some desirable properties, otherwise any kind of a measure between two IT2 FSs could be claimed to be a subsethood measure. Three desirable properties for an IT2 FS subsethood measure $ss(\tilde{A},\tilde{B})$ are (Mendel and Wu 2010, Chap. 4): (1) *Reflexivity*: $ss(\tilde{A},\tilde{B}) = 1 \Leftrightarrow \tilde{A} \le \tilde{B}$ (Note: $\tilde{A} \le \tilde{B}$ if $\bar{\mu}_{\tilde{A}}(x) \le \bar{\mu}_{\tilde{B}}(x)$ and $\underline{\mu}_{\tilde{A}}(x) \le \underline{\mu}_{\tilde{B}}(x)$ for $x \in X$); (2) *Transitivity*: If $\tilde{C} \le \tilde{A} \le \tilde{B}$, then $ss(\tilde{A},\tilde{C}) \ge ss(\tilde{B},\tilde{C})$; or, if $\tilde{A} \le \tilde{B}$, then $ss(\tilde{C},\tilde{A}) \le ss(\tilde{C},\tilde{B})$ for any $\tilde{C}$; and (3) *Overlapping*: If $\tilde{A}\cap\tilde{B} \ne \varnothing$, then $ss(\tilde{A},\tilde{B}) > 0$; otherwise, $ss(\tilde{A},\tilde{B}) = 0$.

[25]A proof that this subsethood measure satisfies all three of the desirable properties for an IT2 FS subsethood measure, that are explained in Footnote 24, can be found in and Mendel and Wu (2010, Appendix 4B.3).

References

Bilgin, A., H. Hagras, A. Malibari, M. Alhaddad, and D. Alghazzawi. 2012a. Towards a general type-2 fuzzy logic approach for computing with words using linear adjectives. In *Proceedings of FUZZ-IEEE 2012*, 1130–1137. Brisbane, AU.

Bilgin, A., J. Dooley, L. Whittington, H. Hagras, M. Henson, C. Wagner, A. Malibari, A. Al-Ghamdi, M. Alhaddad, and D. Alghazzawi. 2012b. Dynamic profile-selection for zslices based type-2 fuzzy agents controlling multi-user ambient intelligent environments. In *Proceedings of FUZZ-IEEE 2012*, 1392–1399. Brisbane, AU.

Bilgin, A., H. Hagras, A. Malibari, M. J. Alhaddad, and D. Alghazzawi. 2012c. A general type-2 fuzzy logic approach for adaptive modeling of perceptions for computing with words. In *Proceedings of 2012 12th UK Workshop on Computational Intelligence* (*UKCI*), 1–8.

Bilgin, A., H. Hagras, A. Malibari, M. J. Alhaddad, and D. Alghazzawi. 2013a. An experience based linear general type-2 fuzzy logic approach for computing with words. In *Proceedings of IEEE Int'l. Conference on Fuzzy Systems*, Paper #1139, Hyderabad, India.

Bilgin, A., H. Hagras, A. Malibari, M. J. Alhaddad, and D. Alghazzawi. 2013b. Towards a linear general type-2 fuzzy logic based approach for computing with words. *Soft Computing*.

Bustince, H., et al. 2016. A historical account of types of fuzzy sets and their relationships. *IEEE Transactions on Fuzzy Systems* 24: 179–194.

Chen, Q., and S. Kawase. 2000. On fuzzy-valued fuzzy reasoning. *Fuzzy Sets and Systems* 113: 237–251.

Cornelis, C., and E. Kerre. 2004. Inclusion measures in intuitionistic fuzzy set theory. *Lecture Notes in Computer Science* 2711: 345–356.

Coupland, S. and R. I. John. (2004) A new and efficient method for the type-2 meet operation. In *Proceedings of IEEE FUZZ Conference*, 959–964. Budapest, Hungary.

Coupland, S., and R. I. John. 2005. Towards more efficient type-2 fuzzy logic systems. In *Proceedings of IEEE FUZZ Conference*, 236–241. Reno, NV, May 2005.

Coupland, S., and R.I. John. 2007. Geometric type-1 and type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 15: 3–15.

Coupland, S., and R. I. John. 2013. Geometric type-2 fuzzy sets. In *Advances in type-2 fuzzy sets and systems: Theory and applications, edited by* A. Sadeghian, J. M. Mendel, and H. Tahayori. Springer, New York.

Dubois, D., and H. Prade. 1978. Operations on fuzzy numbers. *International Journal of Systems Science* 9: 613–626.

Dubois, D., and H. Prade. 1979. Operations in a fuzzy-valued logic. *Information and Control* 43: 224–240.

Dubois, D., and H. Prade. 1980. *Fuzzy sets and systems: Theory and applications*. NY: Academic Press.

Greenfield, S., and R. I. John. 2007. Optimized generalized type-2 join and meet operations. In *Proceeding of FUZZ-IEEE 2007,* 141–146. London, UK.

Hamrawi, H., S. Coupland, and R. John. 2010. A novel alpha-cut representation for type-2 fuzzy sets. In *Proceeding of FUZZ-IEEE 2010*, *IEEE World Congress on Computational Intelligence*, 351–358. Barcelona, Spain.

Hao, M., and J.M. Mendel. 2014. Similarity measures for general type-2 fuzzy sets based on the α-plane representation. *Information Sciences* 277: 197–215.

Harding, J., C. Walker, and E. Walker. 2010. The variety generated by the truth value algebra of type-2 fuzzy sets. *Fuzzy Sets and Systems* 161: 735–749.

John, R., J. Mendel, and J. Carter. 2006. The extended sup-star composition for type-2 fuzzy sets made simple. In *Proceeding of 2006 IEEE International Conference on Fuzzy Systems*, 1441–1445. Vancouver, BC, Canada.

Karnik, N. N., and J. M. Mendel. 1988a. Introduction to type-2 fuzzy logic systems. In *Proceedings of 1998 IEEE FUZZ Conference*, 915–920. Anchorage, AK.

Karnik, N. N., and J. M. Mendel. 1998b. *Operations on Type-2 Fuzzy Sets*. USC-SIPI Report # 319, University of Southern California, Los Angeles, CA. This can be accessed at: http://sipi.usc.edu/research; then choose “sipi technical reports/319.”.

Karnik, N. N., and J. M. Mendel. 1998c. *An Introduction to Type-2 Fuzzy Logic Systems*, USC-SIPI Report #418, University of Southern California, Los Angeles, CA. This can be accessed at: http://sipi.usc.edu/research; then choose “sipi technical reports/418.”.

Karnik, N.N., and J.M. Mendel. 2001. Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems* 122: 327–348.

Klir, G.J., and T.A. Folger. 1988. *Fuggy sets, uncertainity, and information*. Englewood Cliffs, NJ: Prentice-Hall.

Klir, G.J., and B. Yuan. 1995. *Fuzzy sets and fuzzy logic: Theory and applications*. Upper Saddle River, NJ: Prentice Hall.

Kreinovich, V., and G. Xiang. 2008. Towards fast algorithms for processing type-2 fuzzy data: Extending Mendel’s algorithms from interval-valued to a more general case, In *Proceeding of NAFIPS 2008*, Paper #60106, New York City.

Lee, L.-W., and S.-M. Chen. 2008. A new method for fuzzy multiple attributes group decision-making based on the arithmetic of interval type-2 fuzzy sets. In *Proceeding of the Seventh International Conference on Machine Learning and Cybernetics*, 3084–3089. Kunming, China.

Lin, C.-T., and C.S.G. Lee. 1996. *Neural fuzzy systems: A neuro-fuzzy synergism to intelligent systems*. Upper Saddle River, NJ: Prentice-Hall PTR.

Livi, L., H. Tahayori, A. Sadeghian, and A. Rizzi. 2014. Distinguishability of interval type-2 fuzzy sets data by analyzing upper and lower membership functions. *Applied Soft Computing* 17: 79–89.

Mendel, J.M. 2001. *Introduction to rule-based fuzzy logic systems*. Upper Saddle River, NJ: Prentice-Hall.

Mendel, J.M. 2009. On answering the question ‘Where do I start in order to solve a new problem involving interval type-2 fuzzy sets?’. *Information Sciences* 179: 3418–3431.

Mendel, J. M. 2011. On the geometry of join and meet calculations for general type-2 fuzzy sets. In *Proceedings of FUZZ-IEEE 2011*, 2407–2413. Taipei, Taiwan.

Mendel, J.M., R.I. John, and F. Liu. 2006. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems* 14: 808–821.

Mendel, J.M., F. Liu, and D. Zhai. 2009. Alpha-plane representation for type-2 fuzzy sets: Theory and applications. *IEEE Transactions on Fuzzy Systems* 17: 1189–1207.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: Aiding people in making subjective judgments*. Hoboken, NJ: Wiley and IEEE Press.

Miyakoshi, M., Y. Sato, and M. Kawaguchi. 1980. A fuzzy-fuzzy relation and its application to the clustering technique. *Behaviormetrika* 8: 15–22.

Mizumoto, M., and K. Tanaka. 1976. Some properties of fuzzy sets of type-2. *Information and Control* 31: 312–340.

Mizumoto, M., and K. Tanaka. 1981. Fuzzy sets of type-2 under algebraic product and algebraic sum. *Fuzzy Sets and Systems* 5: 277–290.

Nieminen, J. 1977. On the algebraic structure of fuzzy sets of type-2. *Kybernetica*, 13 (4): 261–273.

Nguyen, H. T., and V. Kreinovich. 2008.Computing degrees of subsethood and similarity for interval-valued fuzzy sets: Fast algorithms. In *Proceedings of of 9th International Conference on Intelligent Technologies in Tech’08*, 47–55. Samui, Thailand.

Rajati, M.R., and J.M. Mendel. 2013. Novel weighted averages versus normalized sums in Computing With Words. *Information Sciences* 235: 130–149.

Rickard, J. T., J. Aisbett, G. Gibbon, and D. Morgenthaler. 2008. Fuzzy subsethood for type-n fuzzy sets. In *Proceeding of NAFIPS*, New York.

Ruiz-Garcia, G., H. Hagras, H. Pomares, I. Rojas, and H. Bustince. 2016. Join and meet operations for type-2 fuzzy sets with non-convex secondary memberships. *IEEE Transactions on Fuzzy Systems*.

Tahayori, H., A. G. B. Tettamanzi, and G. D. Antoni. 2006. Approximated type-2 fuzzy set operations. In *Proceedings of FUZZ-IEEE 2006*, 9042–9049. Vancouver, B.C. Canada.

Wagner, C., and H. Hagras. 2008. z Slices—towards bridging the gap between interval and general type-2 fuzzy logic. In *Proceeding of IEEE FUZZ Conference*, Paper # FS0126, Hong Kong, China.

Wagner, C., and H. Hagras. 2010. Towards general type-2 fuzzy logic systems based on zSlices. *IEEE Transactions on Fuzzy Systems* 18: 637–660.

Wagner, C., and H. Hagras. 2013. zSlices based general type-2 fuzzy sets and systems. In *Advances in Type-2 Fuzzy Sets and Systems: Theory and Applications,* edited by A. Sadeghian, J. M. Mendel and H. Tahayori. Springer, New York.

Walker, C.L., and E. Walker. 2005. The algebra of fuzzy truth values. *Fuzzy Sets Systems* 149: 309–347.

Walker, C. L., and E. Walker. 2006. Automorphisms of the algebra of fuzzy truth values. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14: 711–732.

Walker, C.L., and E.A. Walker. 2009. Sets with type-2 operations. *International Journal of Approximate Reasoning* 50: 63–71.

Walker, C.L., and E.A. Walker. 2014. Type-2 operations on finite chains. *Fuzzy Sets and Systems* 236: 33–49.

Wei, S.-H., and S.-M. Chen. 2009. Fuzzy risk analysis based on interval-valued fuzzy numbers. *Expert Systems with Applications* 36: 2285–2299.

Wu, D., and J.M. Mendel. 2007. Uncertainty measures for interval type-2 fuzzy sets. *Information Sciences* 177: 5378–5393.

Wu, D., and J.M. Mendel. 2009. A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets. *Information Sciences* 179: 1169–1192.

Zadeh, L.A. 1975. The concept of a linguistic variable and its application to approximate reasoning–1. *Information Sciences* 8: 199–249.

Zeng, W. Y. Zhou, and H. Li. 2007. Extension principle of interval-valued fuzzy set. In *Fuzzy Information and Engineering: Proceedings of the Second International Conference on Fuzzy Information and Engineering* (*ICFIE*), 125–137. Springer.

Zhai, D., and J.M. Mendel. 2011. Uncertainty measures for general type-2 fuzzy sets. *Information Sciences* 181: 503–518.

Chapter 8
Type-Reduction

8.1 Introduction

One of the major steps in any rule-based fuzzy system is going from a fuzzy set to a number. For a type-1 rule-based fuzzy system, this is done by defuzzification (Sect. 3.6), which may be interpreted as a mapping of a T1 FS into a number. For a type-2 rule-based fuzzy system two avenues are available: (1) map a T2 FS directly into a number—*direct defuzzification*, or (2) first map a T2 FS into a T1 FS (this is called *type-reduction*) and then map that set into a number (defuzzification)—*type-reduction + defuzzification*. Both of these avenues are discussed in great detail in Chaps. 9 and 11.

Some authors refer to direct defuzzification as type-reduction. On the one hand, an argument can be made that direct defuzzification reduces a T2 FS to a type-0 fuzzy set, and so it deserves to be called "type-reduction." On the other hand, historically, the term "type-reduction" means going from a T2 FS to a T1 FS and not directly to a crisp number. Additionally, associating the term "type-reduction" with going directly to a crisp number is inconsistent with using "defuzzification" to do this, because "defuzzification" has been the term used to do this from the very beginning of fuzzy systems. Consequently, in this book direct defuzzification will not be referred to as type-reduction, and this chapter is only about type-reduction, which is a computation that did not exist prior to T2 FSs.

Type-reduction + defuzzification must reduce to defuzzification when the uncertainties of a T2 FS disappear, because of the earlier discussed fundamental design constraint that is adhered to in this book, that: when the membership function (MF) uncertainties of a T2 FS disappear, then a T2 FS must reduce to a T1 FS. This constraint suggests that type-reduction methods should be built upon T1 FS defuzzification methods. Consequently, the type-reduction methods that are presented in this chapter are extensions of type-1 defuzzification methods to T2 FSs.

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_8

The tasks of this chapter are challenging because of the earlier distinction between an interval type-2 fuzzy set (IT2 FS) and a general type-2 fuzzy set (GT2 FS), and the fact that during the past 16 years, type-reduction has been one of the most actively researched topic for T2 FSs and systems, and is now a rather large topic.

The good news is that the horizontal-slice representation for a GT2 FS (which lets a GT2 FS be expressed as the fuzzy union of IT2 FSs each of height α) connects type-reduction for a GT2 FS to type-reduction for an IT2 FS. So, once type-reduction methods have been developed for an IT2 FS, it will be a relatively simple matter to explain how they can be applied to a GT2 FS.

In Chap. 3, centroid, height, and center-of-sets defuzzification were described. All of these methods have one thing in common—each is some sort of a *weighted average* involving crisp numbers. This chapter will demonstrate that each analogous type-reduction method is also a weighted average, but instead of involving crisp numbers each weighted average involves intervals of numbers.

Consequently, the starting point for type-reduction is for IT2 FSs, for which each weighted average is a special kind of *interval weighted average*.

8.2 Interval Weighted Average (IWA)

This section formulates the IWA and provides three popular algorithms for computing it.

8.2.1 Formulation of the IWA

Consider[1] the following arithmetic weighted average[2]:

$$y = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \tag{8.1}$$

In (8.1) let $(i = 1, \ldots, n)$

$$x_i \in [a_i, b_i] \tag{8.2}$$

[1]Some of the material in this section is taken from Mendel and Wu (2010, pp. 147–148), © IEEE 2010.

[2]While it is always true that the sum of the normalized weights in (8.1) add to one, i.e. $\sum_{i=1}^{n} \left(w_i \middle/ \sum_{j=1}^{n} w_j \right) = 1$, it is not a requirement that the sum of the unnormalized weights must add to one.

where x_i may be a positive or negative real number, and

$$w_i \in [c_i, d_i] \tag{8.3}$$

where w_i must be a positive real number, or some but not all w_i may be 0. Sets X_i and W_i, referred to as *intervals*, are associated with (8.2) and (8.3), respectively.

Definition 8.1 When at least one w_i in (8.1) is modeled as in (8.3), and the remaining w_i are modeled as crisp numbers, all of which are subject to the constraints that are stated below (8.3), then the resulting weighted average is called an *interval weighted average* (IWA).

The weighted average in (8.1) is evaluated over the Cartesian product space

$$D_{X_1} \times D_{X_2} \times \cdots \times D_{X_n} \times D_{W_1} \times D_{W_2} \times \cdots \times D_{W_n}.$$

Regardless of the fact that this requires an uncountable number of evaluations,[3] the resulting IWA, Y_{IWA}, will be a closed interval of real numbers, and is completely defined by its two end-points, y_l and y_r, i.e.,[4]:

$$Y_{\mathrm{IWA}} = \frac{\sum_{i=1}^{n} X_i W_i}{\sum_{i=1}^{n} W_i} = [y_l, y_r] \tag{8.4}$$

Because x_i $(i = 1, \ldots, n)$ appear only in the numerator of (8.1), the smallest (largest) value of each x_i is used to find y_l (y_r), i.e.,:

$$y_l = \min_{\forall w_i \in [c_i, d_i]} \frac{\sum_{i=1}^{n} a_i w_i}{\sum_{i=1}^{n} w_i} \tag{8.5}$$

$$y_r = \max_{\forall w_i \in [c_i, d_i]} \frac{\sum_{i=1}^{n} b_i w_i}{\sum_{i=1}^{n} w_i} \tag{8.6}$$

where the notations under min and max in (8.5) and (8.6) mean that i ranges from 1 to n and each w_i ranges from c_i to d_i.

Later in this chapter, the following two cases for x_i are considered: (1) x_i is a crisp real number, so that $a_i = b_i \equiv x_i$, and (2) x_i is an interval of real numbers, so that $a_i \neq b_i$. Because the first case is contained within the second case, the focus here is initially directed at how to compute y_l and y_r as stated in (8.5) and (8.6).

[3]Unless all of the D_{X_i} and D_{W_i} are first discretized, in which case there could still be a very large but countable number of evaluations of (8.1).

[4]In (8.4), $\sum_{i=1}^{n} X_i W_i / \sum_{i=1}^{n} W_i$ is an *expressive* way to summarize the IWA, but it is not a way to compute it.

8.2.2 Computing the IWA

Readers who are familiar with interval arithmetic may suspect that closed-form expressions can be obtained for y_l and y_r. For example, in Klir and Folger (1988) the following formula is given for the division of two interval sets:

$$\begin{aligned}[a,b]/[d,e] &= [a,b] \times [1/e, 1/d] \\ &= [\min(a/d, a/e, b/d, b/e), \max(a/d, a/e, b/d, b/e)]\end{aligned} \tag{8.7}$$

so, it would seem that this result could be applied to determine y_l and y_r. Unfortunately, this cannot be done because the derivation of (8.7) assumes that a, b, d, and e are *independent* (noninteractive). Due to the appearance of w_i in both the numerator and denominator of $\sum_{i=1}^{n} a_i w_i / \sum_{i=1}^{n} w_i$ and $\sum_{i=1}^{n} b_i w_i / \sum_{i=1}^{n} w_i$, the required independence is not present; hence, (8.7) cannot be used to compute the IWA.

Next, let us try calculus. Because $\sum_{i=1}^{n} a_i w_i / \sum_{i=1}^{n} w_i$ and $\sum_{i=1}^{n} b_i w_i / \sum_{i=1}^{n} w_i$ have a similar structure, let

$$y(w_1, \ldots, w_n) \equiv \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \tag{8.8}$$

If the usual calculus approach to optimizing $y(w_1, \ldots, w_n)$ is taken and it is differentiated with respect to any one of the $n\, w_i$, say w_k, it follows that:

$$\begin{aligned}\frac{\partial y(w_1, \ldots, w_n)}{\partial w_k} &= \frac{\partial}{\partial w_k}\left[\frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i}\right] = \frac{\partial}{\partial w_k}\left[\frac{x_k w_k + \sum_{i=1, i\neq k}^{n} x_i w_i}{w_k + \sum_{i=1, i\neq k}^{n} w_i}\right] \\ &= \frac{x_k\left(w_k + \sum_{i=1, i\neq k}^{n} w_i\right) - \left(x_k w_k + \sum_{i=1, l\neq k}^{n} x_i w_i\right) \times 1}{\left(w_k + \sum_{i=1, i\neq k}^{n} w_i\right)^2} \\ &= \frac{x_k}{\sum_{i=1}^{n} w_i} - \frac{\sum_{i=1}^{n} x_i w_i}{\left(\sum_{i=1}^{n} w_i\right)^2} = \frac{x_k}{\sum_{i=1}^{n} w_i} - \frac{\sum_{l=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \times \frac{1}{\sum_{i=1}^{n} w_i} \\ &= \frac{x_k - y(w_1, \ldots, w_n)}{\sum_{i=1}^{n} w_i}\end{aligned} \tag{8.9}$$

Equating $\partial y / \partial w_k$ to zero, and using (8.8), one finds:

$$\begin{aligned}(y(w_1, \ldots, w_n) = x_k) &\Rightarrow \left(\frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} = x_k\right) \Rightarrow \left(\sum_{i=1}^{n} x_i w_i = x_k \sum_{i=1}^{n} w_i\right) \\ &\Rightarrow \left(\sum_{i=1, i\neq k}^{n} x_i w_i = x_k \sum_{i=1, i\neq k}^{n} w_i\right)\end{aligned} \tag{8.10}$$

Observe that w_k no longer appears in the final expression in (8.10), so that the direct calculus approach does not work.

Because[5] $\sum_{i=1}^{n} w_i > 0$, it is easy to see from (8.9) that

$$\frac{\partial y(w_1,\ldots,w_n)}{\partial w_k}\begin{cases}\geq 0 & \text{if } x_k \geq y(w_1,\ldots,w_n)\\ <0 & \text{if } x_k < y(w_1,\ldots,w_n)\end{cases} \tag{8.11}$$

This equation gives the directions in which w_k should be changed so as to either increase or decrease $y(w_1,\ldots,w_n)$, which will lead to an iterative approach for computing the IWA, i.e.,:

$$\begin{cases}\text{If } x_k > y(w_1,\ldots,w_n) & \\ \qquad y(w_1,\ldots,w_n) \text{ increases (decreases) as } w_k \text{ increases (decreases)} & \\ \text{If } x_k < y(w_1,\ldots,w_n) & \\ \qquad y(w_1,\ldots,w_n) \text{ increases (decreases) as } w_k \text{ decreases (increases)} & \end{cases} \tag{8.12}$$

Observe, from (8.3), that the maximum value that w_k can attain is d_k and the minimum value that it can attain is c_k. Consequently, (8.12) implies that $y(w_1,\ldots,w_n)$ *attains its minimum value*, y_l, if [focus on $y(w_1,\ldots,w_n)$ "decreases" in each line of (8.12)]:

$$w_k = \begin{cases} c_k & \forall k \text{ such that } x_k > y(w_1,\ldots,w_n)\\ d_k & \forall k \text{ such that } x_k < y(w_1,\ldots,w_n)\end{cases} \tag{8.13}$$

Similarly, it can be deduced from (8.12) that $y(w_1,\ldots,w_n)$ *attains its maximum value*, y_r, if [focus on $y(w_1,\ldots,w_n)$ "increases" in each line of (8.12)]:

$$w_k = \begin{cases} d_k & \forall k \text{ such that } x_k > y(w_1,\ldots,w_n)\\ c_k & \forall k \text{ such that } x_k < y(w_1,\ldots,w_n)\end{cases} \tag{8.14}$$

Because, there are only two possible choices for w_k in (8.13) or (8.14), to compute y_l (y_r) w_k *switches only one time* between c_k and d_k; hence, y_l and y_r in (8.5) and (8.6) are given by [for y_l, x_k in (8.13) is a_k, whereas for y_r, x_k in (8.14) is b_k]:

$$y_l = y_l(L) = \frac{\sum_{i=1}^{L} a_i d_i + \sum_{i=L+1}^{n} a_i c_i}{\sum_{i=1}^{L} d_i + \sum_{i=L+1}^{n} c_i} \tag{8.15}$$

$$y_r = y_r(R) = \frac{\sum_{i=1}^{R} b_i c_i + \sum_{i=R+1}^{n} b_i d_i}{\sum_{i=1}^{R} c_i + \sum_{i=R+1}^{n} d_i} \tag{8.16}$$

[5]This is where the constraint that w_i must be a positive real number, or some but not all w_i may be 0, is needed.

In (8.15) and (8.16), a_i and b_i $(i = 1, \ldots, n)$ have been put in increasing orders, i.e., $a_1 \le a_2 \le \cdots \le a_n$ and $b_1 \le b_2 \le \cdots \le b_n$ [so that it is easy to perform the tests in (8.13) and (8.14)], L and R are called[6] *switch points*, and it is these switch points that remain to be determined. In (8.15), when $i = L + 1$, observe that w_i switches from the right-end-points of $[c_i, d_i]$ to its left-end-points, whereas in (8.16) just the opposite occurs.

Wu (2011) has provided the following insights about the structure of (8.15) [(8.16)] as it relates to the solution of the optimization problem in (8.5) [(8.6)]: Examining the right-hand side of (8.5), in order to obtain the smallest value of $\sum_{i=1}^{n} a_i w_i \big/ \sum_{i=1}^{n} w_i$ one wants to associate the largest value of w_i (namely, d_i) with the smallest values of a_i and then the smallest values of w_i (namely, c_i) with the largest values of a_i. The opposite is true for obtaining the largest value of $\sum_{i=1}^{n} b_i w_i \big/ \sum_{i=1}^{n} w_i$ in (8.6).

In general $R \ne L$, and no closed-form formulas are known for L and R [except in one very special case (see Example 8.13)]. Instead, each is computed by means of an iterative algorithm.

Before turning to such algorithms, it is useful to express (8.15) and (8.16) in the following alternative ways (Liu and Mendel 2011; Wu and Mendel 2007b):

$$y_l = \min_{k=1,2,\ldots,n} y_l(k) = \min_{k=1,2,\ldots,n} \frac{\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i}{\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i} \tag{8.17}$$

or

$$\begin{cases} y_l = \dfrac{\sum_{i=1}^{n} a_i \left\{ \delta_l^i d_i + (1-\delta_l^i) c_i \right\}}{\sum_{i=1}^{n} \left\{ \delta_l^i d_i + (1-\delta_l^i) c_i \right\}} \\ \delta_l^i = \begin{cases} 1 & \text{if } a_i \le y_l \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{8.18}$$

and

$$y_r = \max_{k=1,2,\ldots,n} y_r(k) = \max_{k=1,2,\ldots,n} \frac{\sum_{i=1}^{k} b_i c_i + \sum_{i=k+1}^{k} b_i d_i}{\sum_{i=1}^{k} b_i c_i + \sum_{i=k+1}^{k} b_i d_i} \tag{8.19}$$

or

$$\begin{cases} y_r = \dfrac{\sum_{i=1}^{k} b_i \left\{ \delta_r^i d_i + (1-\delta_r^i) c_i \right\}}{\sum_{i=1}^{n} \left\{ \delta_r^i d_i + (1-\delta_r^i) c_i \right\}} \\ \delta_r^i = \begin{cases} 1 & \text{if } b_i \ge y_r \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{8.20}$$

[6] Actually, it is a_L and b_R that are the switch points, for which L and R are the corresponding indices. Because calling L and R "switch points" is so entrenched in the type-2 literature, this book continues to use the same terminology.

In (8.17) and (8.19), each value of k is a *candidate switch point*, a_k (or b_k) is associated with it, and a_i, b_i $(i = 1, \ldots, n)$ are in increasing orders. On the other hand, in (8.19) and (8.20) a_i, b_i $(i = 1, \ldots, n)$ are in their natural (nonincreasing) orders, something that is very useful when derivatives of y_l and y_r have to be computed.

A *brute-force* way to find y_l (y_r) is to compute $\{y_l(k)\}_{k=1}^{n-1}$ [$\{y_r(k)\}_{k=1}^{n-1}$] and then locate its smallest (largest) element. This is called "brute force" (or exhaustive) because it does not focus on a good way to initialize the process, move from one step to the next, or stop the process. All algorithms devised for finding y_l (y_r) focus on at least one of these three requirements of an optimization algorithm.

8.2.3 KM Algorithms

Many iterative algorithms have been developed for computing L and R, and subsequently y_l and y_r. The first such algorithms were developed in Karnik and Mendel (2001), and are now known as *KM algorithms*. They are still the most widely used algorithms for computing y_l and y_r, and are given in Table 8.1.

At the start of each KM algorithm, all $a_i(b_i)$ must be rank ordered, in increasing order. Doing this means that, after this rank ordering, all w_i have to be relabeled so that they correspond to the rank-ordered a_i and b_i; and, the relabeling for the rank-ordered a_i will usually be different than the relabeling for the rank-ordered b_i, because b_i are usually not ranked in the same order as a_i. It is only when $a_i =$

Table 8.1 KM algorithms for computing the end-points of an IWA

Step	KM algorithm for $y_l(L)$ $y_l(L) = \min\limits_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} a_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$	KM algorithm for $y_r(R)$ $y_r(R) = \max\limits_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} b_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$
1	Initialize w_i by setting $w_i = (c_i + d_i)/2$, $i = 1, \ldots, n$, and then compute $y' = y(w_1, \ldots, w_n) = \sum_{i=1}^{n} a_i w_i \Big/ \sum_{i=1}^{n} w_i$	$y' = y(w_1, \ldots, w_n) = \sum_{i=1}^{n} b_i w_i \Big/ \sum_{i=1}^{n} w_i$
2	Find $k \in \{1, 2, \ldots, n-1\}$ such that $a_k \le y' \le a_{k+1}$	Find $k \in \{1, 2, \ldots, n-1\}$ such that $b_k \le y' \le b_{k+1}$
3	Set $w_i = d_i$ when $i \le k$, and $w_i = c_i$ when $i \ge k+1$, and then compute $y_l(k) \equiv \dfrac{\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{k} a_i c_i}{\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{k} c_i}$	Set $w_i = c_i$ when $i \le k$, and $w_i = d_i$ when $i \ge k+1$, and then compute $y_r(k) = \dfrac{\sum_{i=1}^{k} b_i c_i + \sum_{i=k+1}^{n} b_i d_i}{\sum_{i=1}^{k} c_i + \sum_{i=k+1}^{n} d_i}$
4	Check if $y_l(k) = y'$. If yes, stop and set $y_l(k) = y_l(L)$ and call k L. If no, go to Step 5	Check if $y_r(k) = y'$. If yes, stop and set $y_r(k) = y_r(R)$ and call k R. If no, go to Step 5
5	Set $y' = y_l(k)$ and go to Step 2	Set $y' = y_r(k)$ and go to Step 2

Note that $a_1 \le a_2 \le \cdots \le a_n$ and $b_1 \le b_2 \le \cdots \le b_n$ [adapted from Mendel and Wu (2010, Chap. 2); © 2010, IEEE]

$b_i \equiv x_i$ and x_i corresponds to a sampled value of a primary variable (in which case x_i are naturally rank-ordered) that x_i are already in a rank ordering.

Example 8.1 The purpose of this example is to illustrate the steps of the KM algorithms. To do this, the starting point is the following data [in this example, $a_i = b_i \equiv x_i \ (i = 1, \ldots, 5)$]:

i	x_i	c_i	d_i	Initial w_i
1	1	0.60	0.90	0.750
2	2	0.50	0.70	0.600
3	3	0.65	0.80	0.725
4	4	0.20	0.40	0.300
5	5	0.30	0.75	0.525

In step 1, w_i are initialized as shown in the last column of this data table, so that $y' = \sum_{i=1}^{n} x_i w_i / \sum_{i=1}^{n} w_i = 2.741$. y_l is computed by its KM algorithm, as shown in Table 8.2, and y_r is computed by its KM algorithm, as shown in Table 8.3.

Observe from the computations in Tables 8.2 and 8.3 that only two iterations are needed to find L, three iterations are needed to find R, and both of these iteration counts are less than the maximum number of averages, which is five. Observe, also, that convergence of the two solutions is not only fast but it occurs monotonically. These observations from simulations suggest that there should be theoretical underpinnings for them. There are, and they are stated and proved in Sect. 2.1 of Appendix 2.

Note that it is only when a_i and b_i are *rank ordered* that one switch point occurs in (8.15) and (8.16). After L and R have been computed and the a_i and b_i are put back into their original ordering, it is quite common for there to be more than one switch among the c_i and d_i, as is illustrated in the next example.

Example 8.2 Figure 8.1a, b depict $x_i \in [a_i, b_i]$ and $w_i \in [c_i, d_i] \ (i = 1, \ldots, 9)$, in which index i is called "Rule number," an index that has no natural ordering.[7]

The indices of the rank ordered a_i and b_i for the computations of $y_l(L)$ and $y_r(R)$ are given in Table 8.4. The ordered values were obtained directly from Fig. 8.1a, on which the lowest (highest) value for each X_i corresponds to a_i (b_i). In order to compute $y_l(L)$, the unordered W_i in Fig. 8.1b must also be ordered according to the ordering schedule given in Table 8.4 for a_i. Figure 8.2a, b depict the ordered X_i (observe the nondecreasing order of the *bottom* of each vertical bar) and W_i. The numbers that are at the top of each vertical bar in both figures denote the original rule number. The black cross-marks in Fig. 8.2b denote which end of W_i was used to compute $y_l(L)$, as determined by its KM algorithm. The red cross-mark is where a switch occurs from the d_i to the c_i, and from it observe that $L = 4$. Although the

[7] The numbering of rules is arbitrary, and their renumbering does not change any computations; hence, there is no natural ordering for them.

Table 8.2 Computation of y_l by its KM algorithm, for Example 8.1

Iteration	Step 2	Step 3	Step 4	Step 5
1	$2 \leq 2.741 \leq 3 \rightarrow k = 2$	$y_l(2) = \frac{1 \times 0.9 + 2 \times 0.7 + 3 \times 0.65 + 4 \times 0.2 + 5 \times 0.3}{0..9 + 0.7 + 0.65 + 0.2 + 0.3} = 2.382$	$y_l(2) \neq y'$	$y' = 2.382$
2	$2 \leq 2.382 \leq 3 \rightarrow k = 2$	$y_l(2) = \frac{1 \times 0.9 + 2 \times 0.7 + 3 \times 0.65 + 4 \times 0.2 + 5 \times 0.3}{0.9 + 0.7 + 0.65 + 0.2 + 0.3} = 2.382$	$y_l(2) = y' = 2.382$ STOP $L = 2$	

Table 8.3 Computation of y_r by its KM algorithm, for Example 8.1

Iteration	Step 2	Step 3	Step 4	Step 5
1	$2 \leq 2.741 \leq 3 \rightarrow k = 2$	$y_r(2) = \frac{1 \times 0.6 + 2 \times 0.5 + 3 \times 0.8 + 4 \times 0.4 + 5 \times 0.75}{0.6 + 0.5 + 0.8 + 0.4 + 0.75} = 3.066$	$y_r(2) \neq y'$	$y' = 3.066$
2	$3 \leq 3.066 \leq 4 \rightarrow k = 3$	$y_r(3) = \frac{1 \times 0.6 + 2 \times 0.5 + 3 \times 0.65 + 4 \times 0.4 + 5 \times 0.75}{0.6 + 0.5 + 0.65 + 0.4 + 0.75} = 3.069$	$y_r(3) \neq y' = 3.066$	$y' = 3.069$
3	$3 \leq 3.069 \leq 4 \rightarrow k = 3$	$y_r(3) = \frac{1 \times 0.6 + 2 \times 0.5 + 3 \times 0.65 + 4 \times 0.4 + 5 \times 0.75}{0.6 + 0.5 + 0.65 + 0.4 + 0.75} = 3.069$	$y_r(3) = y' = 3.069$ STOP $R = 3$	

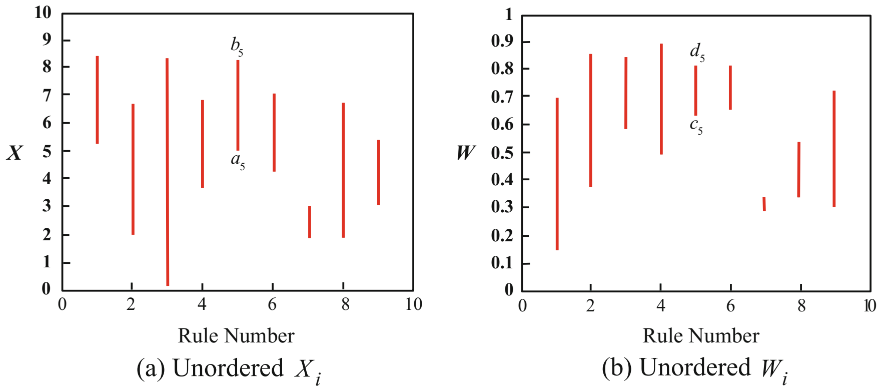


Fig. 8.1 Unordered **a** X_i and **b** W_i for Example 8.2

Table 8.4 Indices of the rank ordered a_i and b_i for the computations of $y_l(L)$ and $y_r(R)$ in Example 8.2

a_i		b_i	
i_{original}	i_{ordered}	i_{original}	i_{ordered}
1	9	1	9
2	4	2	3
3	1	3	8
4	6	4	5
5	8	5	7
6	7	6	6
7	2	7	1
8	3	8	4
9	5	9	2

details of the IWA calculations are not provided here for $y_l(L)$, its computed value is $y_l(L) = 3.02$. Figure 8.2c, obtained from Fig. 8.2b, depicts the unordered W_i for y_l (as in Fig. 8.1b). Observe that there are four switches from c_i to d_i, or vice versa, that occur at i_{original} = 1, 3, 6, and 8.

Similarly, in order to compute $y_r(R)$, the unordered W_i in Fig. 8.1b must also be ordered according to the ordering schedule given in Table 8.4, but this time for b_i. Figure 8.3a, b depict the ordered X_i (observe the nondecreasing order of the *top* of each vertical bar) and W_i. Again, the numbers that are at the top of each vertical bar denote the original rule number; the black cross-marks in Fig. 8.3b denote which end of W_i was used to compute $y_r(R)$ as determined by its KM algorithm; and, the red cross-mark is where a switch occurs from the c_i to the d_i, and from it observe that $R = 6$. Although the details of the IWA are not provided here for $y_r(R)$, its computed value is $y_r(R) = 7.17$. Figure 8.3c, obtained from Fig. 8.3b, depicts the unordered W_i for y_r (as in Fig. 8.1b). Observe that there are five switches from d_i to c_i, or vice versa, that occur at i_{original} = 1, 2, 3, 4, and 5.

Comparing Figs. 8.2c and 8.3c, observe that it is possible for some (but not all) of the same value of W_i to be used in the calculations of both y_l and y_r.

In summary, for this example, $Y_{\text{IWA}} = [y_l, y_r] = [3.02, 7.17]$.

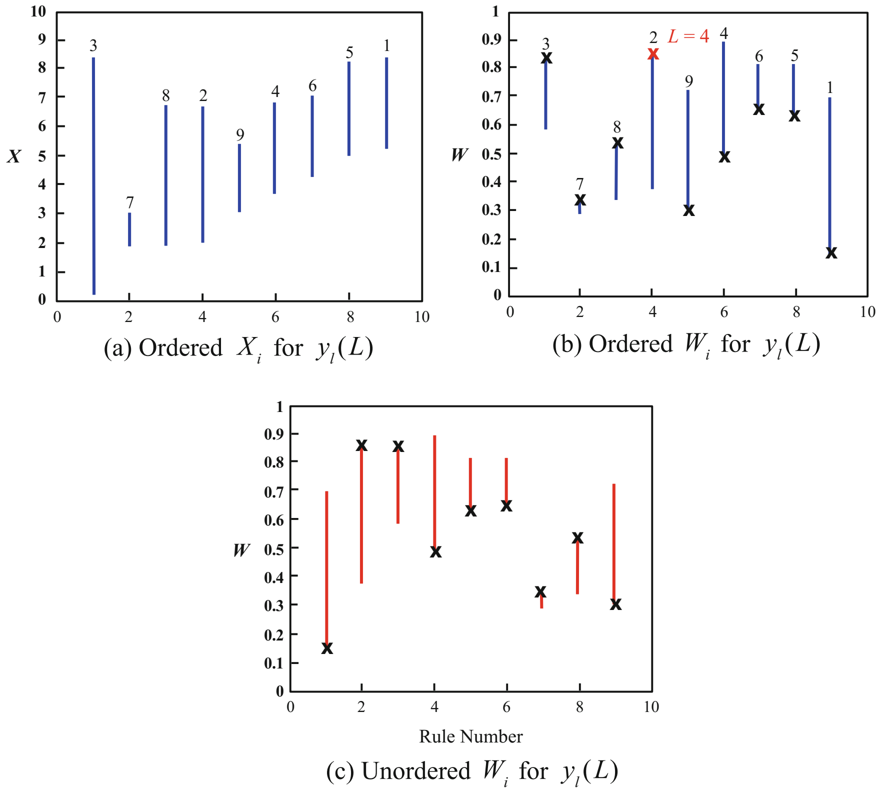


Fig. 8.2 Ordered **a** X_i and **b** W_i for computing $y_l(L)$, and **c** unordered W_i for computing $y_l(L)$. In **a** and **b**, the *number* that is at the *top of each vertical bar* denotes the original rule number

8.2.4 Enhanced KM Algorithms

Recall that any optimization algorithm requires a good way to (1) initialize it, (2) move from one step to the next, and (3) stop. KM algorithms do not provide for a good way for their initialization, nor do they provide the best way for stopping them, but they do provide a good way to move from one step to the next.

The enhanced KM (EKM) algorithms (Wu and Mendel 2009) start with the KM algorithms and modify them in three ways: (1) A better initialization is used to reduce the number of iterations; (2) the termination condition of the iterations is changed to remove an unnecessary iteration; and (3) a subtle computing technique is used to reduce the computational cost of each algorithm's iterations. The EKM algorithms are summarized in Table 8.5.

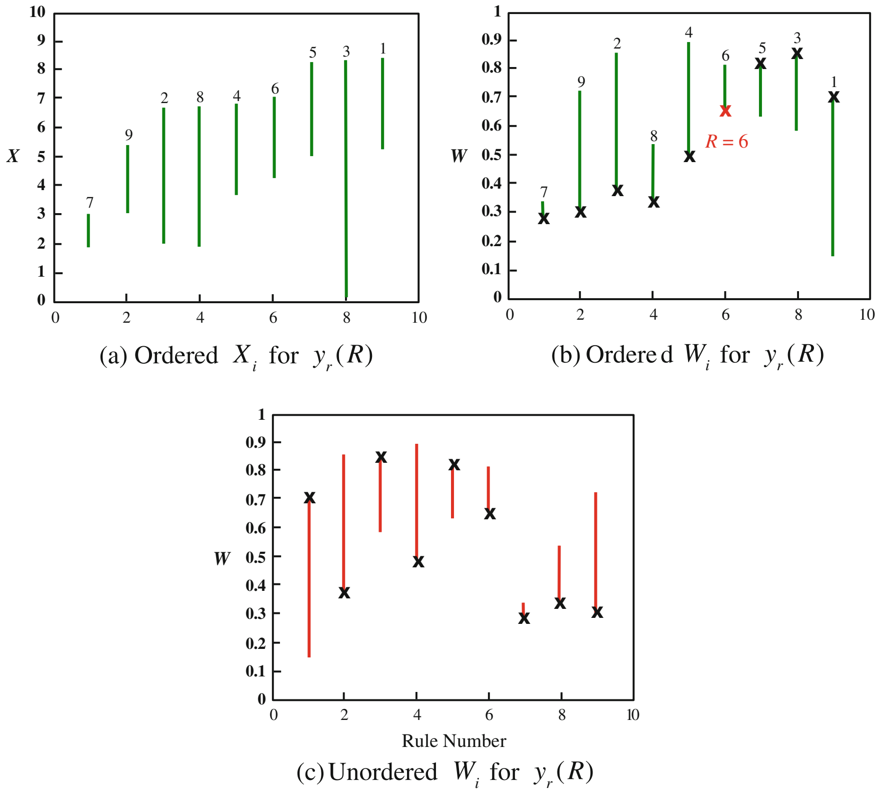


Fig. 8.3 Ordered **a** X_i and **b** W_i for computing $y_r(R)$, and **c** unordered W_i for computing $y_r(R)$. In **a** and **b**, the *number* that is at the *top of each vertical bar* denotes the original rule number

The better initializations are shown in Step 1 of Table 8.5, and both were obtained from extensive simulations.[8] A close examination of Steps 2–5 in Table 8.1 reveals that the termination conditions can be moved one step earlier, something that is done in Table 8.5. The "subtle computing technique" uses the fact that very little changes from one iteration to the next, so instead of re-computing everything on the right-hand sides of y', as is done in Step 3 of Table 8.1, only the portions of those right-hand sides that do change are recomputed, as is done in Step 4 of Table 8.5.

Extensive simulations have shown that on average the EKM algorithms can save about two iterations, which corresponds to a more than 39% reduction in computation time.

---

[8]The two Table 8.5 EKM algorithm initializations were obtained by Dongrui Wu using massive Monte Carlo simulations. They were derived mathematically in Liu et al. (2012a). Recently, Salaken et al. (2016) showed, also by means of massive Monte Carlo simulations, that setting $k = [n/1.6]$ to initialize the EKM algorithm for y_r is even better than using $k = [n/1.7]$, whereas no change is needed in the initialization of the EKM algorithm for y_l (see Sect. 8.2.6 for additional discussions).

Table 8.5 EKM algorithms for computing the end-points of an IWA

Step	EKM Algorithm for $y_l(L)$ $y_l(L) = \min_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} a_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$	EKM Algorithm for $y_r(R)$ $y_r(R) = \max_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} b_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$
1	Set $k = [n/2.4]$ (the nearest integer to $n/2.4$) and compute: $a = \sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i$ $b = \sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i$ Compute $y' = a/b$	Set $k = [n/1.7]$ (the nearest integer to $n/1.7$) and compute $a = \sum_{i=1}^{k} b_i c_i + \sum_{i=k+1}^{n} b_i d_i$ $b = \sum_{i=1}^{k} c_i + \sum_{i=k+1}^{n} d_i$ Compute $y' = a/b$
2	Find $k' \in \{1, 2, \ldots, n-1\}$ such that $a_{k'} \le y' \le a_{k'+1}$	Find $k' \in \{1, 2, \ldots, n-1\}$ such that $b_{k'} \le y' \le b_{k'+1}$
3	Check if $k' = k$. If yes, stop and set $y' = y_l(L)$, and $k = L$. If no, go to Step 4	Check if $k' = k$. If yes, stop and set $y' = y_r(R)$, and $k = R$. If no, go to Step 4
4	Compute $s = sign(k' - k)$ and: $a' = a + s \sum_{i=\min(k,k')+1}^{\max(k,k')} a_i(d_i - c_i)$ $b' = b + s \sum_{i=\min(k,k')+1}^{\max(k,k')} (d_i - c_i)$ Compute $y''(k') = a'/b'$	Compute $s = sign(k' - k)$ and: $a' = a - s \sum_{i=\min(k,k')+1}^{\max(k,k')} b_i(d_i - c_i)$ $b' = b - s \sum_{i=\min(k,k')+1}^{\max(k,k')} (d_i - c_i)$ Compute $y''(k') = a'/b'$
5	Set $y' = y''(k')$, $a = a'$, $b = b'$ and $k = k'$ and go to Step 2	

Note that $a_1 \le a_2 \le \cdots \le a_n$ and $b_1 \le b_2 \le \cdots \le b_n$ [adapted from Mendel and Wu (2010, Chap. 2); © 2010, IEEE]

Example 8.3 The purpose of this example is to illustrate the steps of the EKM algorithm and to then compare the number of EKM algorithm iterations with the number of KM algorithm iterations. To do this, the starting point is the same data that are used in Example 8.1 for which $a_i = b_i \equiv x_i \ (i = 1, \ldots, 5)$.

y_l is computed by its EKM Algorithm as follows: In Step 1, $k = 5/2.4 = [2.08] \to 2$ so that:

$$y' = \frac{1 \times 0.9 + 2 \times 0.7 + 3 \times 0.65 + 4 \times 0.2 + 5 \times 0.3}{0.9 + 0.7 + 0.65 + 0.2 + 0.3} = 2.382.$$

Iterations begin with Step 2, for which: $2 \le 2.382 \le 3 \to k' = 2$. In Step 3, $k' = k$ and so the algorithm stops and sets $L = 2$, which agrees with what was obtained in Example 8.1 when the KM algorithm was used.

y_r is computed by its EKM Algorithm as follows: In Step 1, $k = 5/1.7 = [2.94] \to 3$ so that:

$$y' = \frac{1 \times 0.6 + 2 \times 0.5 + 3 \times 0.65 + 4 \times 0.4 + 5 \times 0.75}{0.6 + 0.5 + 0.65 + 0.4 + 0.75} = 3.069.$$

Iterations again begin with Step 2, for which: $3 \leq 3.069 \leq 4 \to k = 3$. In Step 3, $k' = k$ and so the algorithm stops and sets $R = 3$, which agrees with what was obtained in Example 8.1 when the KM algorithm was used.

Observe that only one iteration was needed to find L and R, and, as compared to the number of iterations used by the KM Algorithms, this represents a 50% savings in iterations for the calculation of L and a 67% savings in iterations for the calculation of R. Of course, this is a toy example, but it does indicate that savings in iterations are possible when the EKM algorithms are used.

It is important for the reader to appreciate that the word "enhanced" in "EKM" is a synonym for "better," which means that *one should no longer use KM algorithms, and should instead use EKM algorithms*. This is mentioned because many papers still show results for both the KM and EKM algorithms, which this author feels is unnecessary.

8.2.5 Enhanced Iterative Algorithm with Stopping Condition (EIASC)[9]

At the end of Sect. 8.2.2 a brute-force algorithm was mentioned that did not satisfy any of the three desired requirements of an optimization algorithm [a good way to (1) initialize it, (2) move from one step to the next, and (3) stop]. Melgerejo (2007) and Duran et al. (2008) have beefed up the brute-force algorithm in their *iterative algorithm + stopping condition* (IASC), and Wu and Nie (2011) have made some improvements to it in their *enhanced IASC* (EIASC), which is given in Table 8.6.

The EIASC algorithm for y_l begins by using only the left-ends of $W_i = [c_i, d_i]$, and, unless the stopping condition is satisfied, it replaces one c_i (beginning with c_1) with its respective d_i (beginning with d_1) during each successive iteration, until the stopping condition is satisfied. The stopping condition in Step 3 is a result of the Location Property for $y_l(L)$ that is explained in Appendix 2.

Similarly, the EIASC algorithm for y_r begins by using only the right-ends of $W_i = [c_i, d_i]$, and, unless the stopping condition is satisfied, it replaces one d_i (beginning with d_n) with its respective c_i (beginning with c_n) during each successive iteration, until the stopping condition is satisfied. The stopping condition in Step 3 is also a result of the Location Property for $y_r(R)$ that is explained in Appendix 2.

[9]Except for Example 8.4, the material in this section is taken from Mendel (2013, p. 436).

Table 8.6 EIASC for computing the end-points of an IWA

Step	EIASC for $y_l(L)$ $y_l(L) = \min_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} a_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$	EIASC for $y_r(R)$ $y_r(R) = \max_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} b_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$
1	Initialize $a = \sum_{i=1}^{n} a_i c_i \; b = \sum_{i=1}^{n} c_i$ $L = 0$	Initialize $a = \sum_{i=1}^{n} b_i d_i \; b = \sum_{i=1}^{n} d_i$ $R = n$
2	Compute $L = L + 1$ $a = a + a_L(d_L - c_L)$ $b = b + (d_L - c_L)$ $y_l(L) = a/b$	Compute $a = a + b_R(d_R - c_R)$ $b = b + (d_R - c_R)$ $y_r(R) = a/b$ $R = R - 1$
3	If $y_l(L) \leq a_{L+1}$, stop, otherwise go to Step 2	If $y_r(R+1) \geq b_R$, stop, otherwise go to Step 2

Note that $a_1 \leq a_2 \leq \cdots \leq a_n$ and $b_1 \leq b_2 \leq \cdots \leq b_n$ (Wu and Nie 2011)

One could ask: Why was not each EIASC algorithm initialized as in the respective EKM algorithm (see Step 1 in Table 8.5)? Wu (2011) actually tried this, but his simulations showed that this further enhanced EIASC algorithm only outperformed the EIASC algorithm when $n > 1000$. When the IWA is used for type-reduction in a type-2 fuzzy system (in their case, a fuzzy controller) usually $n \ll 1000$; hence, this further enhanced EIASC algorithm was not recommended.

Example 8.4 The purpose of this example is to illustrate the steps of the EIASC algorithm and to then compare the number of EIASC algorithm iterations with the number of KM and EKM algorithm iterations. To do this, the starting point is the same data that are used in Examples 8.1 and 8.3 for which $(i = 1, \ldots, 5)$ $a_i = b_i \equiv x_i = 1, 2, 3, 4, 5$, and, c_i and d_i are in the Example 8.1 table.

y_l is computed by its EIASC Algorithm as follows: a and b are initialized as

$$a = 1 \times 0.6 + 2 \times 0.5 + 3 \times 0.65 + 4 \times 0.2 + 5 \times 0.3 = 5.85$$

and

$$b = 0.6 + 0.5 + 0.65 + 0.2 + 0.3 = 2.25,$$

and L is set equal to 0. Iterations begin with Step 2 and are given in Table 8.7.

y_r is computed by its EIASC Algorithm as follows: a and b are initialized as

$$a = 1 \times 0.9 + 2 \times 0.7 + 3 \times 0.8 + 4 \times 0.4 + 5 \times 0.75 = 10.05$$

and

$$b = 0.9 + 0.7 + 0.8 + 0.4 + 0.75 = 3.55,$$

and R is set equal to 5. Iterations begin with Step 2 and are given in Table 8.8.

Table 8.7 Computation of y_l by its EIASC algorithm, for Example 8.4

Iteration	Step 2	Step 3
1	$L = 1$ $a = 5.85 + x_1(d_1 - c_1) = 5.85 + 1 \times (0.9 - 0.6)$ $= 5.85 + 0.3 = 6.15$ $b = 2.25 + (d_1 - c_1) = 2.25 + (0.9 - 0.6)$ $= 2.25 + 0.3 = 2.55$ $y_l(1) = 6.15/2.55 = 2.41$	$y_1(1) \nleq x_2 = 2$
2	$L = 2$ $a = 6.15 + x_2(d_2 - c_2) = 6.15 + 2 \times (0.7 - 0.5)$ $= 6.15 + 0.4 = 6.55$ $b = 2.55 + (d_2 - c_2) = 2.55 + (0.7 - 0.5)$ $= 2.55 + 0.2 = 2.75$ $y_l(2) = 6.55/2.75 = 2.38$	$y_l(2) \leq x_3 = 3$ STOP $L = 2$ (in agreement with L in Examples 8.1 and 8.3)

Observe, from Tables 8.7 and 8.8, that two iterations are needed to find L and R, making it less efficient than EKM for both L and R, equally efficient than KM for L and more efficient than KM for R.

Example 8.5 Wu and Nie (2011) compare the performance of five type-reduction algorithms including KM, EKM, and EIASC. In March 2016, Dongrui Wu reran the simulations in that paper and provided me with the results that are in Fig. 8.4. His platform was a Lenovo Thinkpad T440P with Intel Core i5-4300M CPU @2.60 GHz and 12 GB memory, running 64-bit Windows 10 and MATLAB® R2012a (Matlab is a registered trademark of The MathWorks). Computational costs for each algorithm were measured by the computation time obtained from MATLAB® *tic* and *toc* functions.

Table 8.8 Computation of y_r by its EIASC algorithm, for Example 8.4

Iteration	Step 2	Step 3
1	$a = 10.05 + x_5(d_5 - c_5) = 10.05 + 5 \times (0.75 - 0.3)$ $= 10.05 + 2.25 = 12.3$ $b = 3.55 + (d_5 - c_5) = 3.55 + (0.75 - 0.3)$ $= 3.55 + 0.45 = 4$ $y_r(5) = 12.3/4 = 3.1$ $R = 4$	$y_r(5) \ngeq x_4 = 4$
2	$a = 12.3 + x_4(d_4 - c_4) = 12.3 + 4 \times (0.4 - 0.2)$ $= 12.3 + 0.8 = 13.1$ $b = 4 + (d_4 - c_4) = 4 + (0.4 - 0.2)$ $= 4 + 0.2 = 4.2$ $y_r(4) = 13.1/4.2 = 3.12$ $R = 3$ (in agreement with R in Examples 8.1 and 8.3)	$y_r(4) \geq x_3 = 3$ STOP

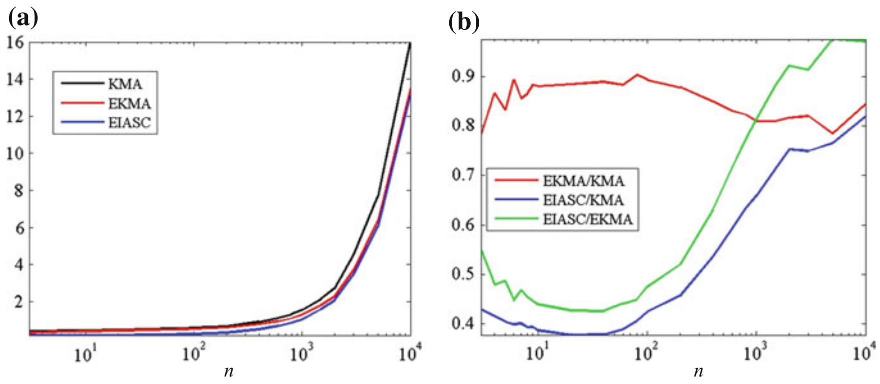


Fig. 8.4 Example 8.5 **a** total computation time (seconds) for the 5000 Monte Carlo simulations for different values of n, and **b** ratio of the computation time of EKM to KM, EIASC to KM and EIASC to EKM. Note that the "A" at the end of each algorithm (e.g., KMA) denotes "Algorithm." (Courtesy of D. Wu, March 2016.)

In his simulations:

- $a_i = b_i \equiv x_i$
- x_i, c_i and d_i were randomly chosen by assuming X_i and W_i to be uniformly distributed over $[0, 1]$.
- n = {3, 4, …,10, 20, 40, …, 100, 200, 400, …, 1000, 1500, 2000, 3000, 5000, 10000}.
- For each n, 5000 Monte Carlo simulations were used to compute Y_{IWA} by each algorithm.

Figure 8.4a depicts the total computation time (seconds) for the 5000 Monte Carlo simulations for different values of n. Figure 8.4b depicts the ratio of the computation time of EKM to KM, EIASC to KM and, most importantly, EAISC to EKM. Observe that the EIASC algorithms outperform the EKM algorithms, for $n < 1300$. The computational cost of EIASC increases rapidly as n increases since it needs to evaluate many switch points before finding the correct ones.

Although the results in Example 8.4 may seem to be a counterexample to this example, they are for only one value of n, x_i, c_i, and d_i, and so one should not rush to draw general conclusions from them, whereas one is able to draw general conclusions from the results in Fig. 8.4, and so they should be abided by. In real-world applications, in which n corresponds to the number of rules, use the EIASC.

8.2.6 Remarks

For readers who are interested in learning more about the two optimization problems that are in (8.5) and (8.6), see Sect. 2.1 in Appendix 2. It provides important properties about these problems, and includes their proofs.

When an iterative algorithm is used to solve an optimization problem, it is important to know whether or not it converges to the correct solution. It should be clear from the statements of the KM, EKM and EIASC algorithms that they all converge to the correct solution (i.e., the global minimum or maximum) and this occurs in a finite number of iterations.

The earliest KM paper (Karnik and Mendel 2001) proved that each KM algorithm requires at most n iterations. In Liu and Mendel (2008, Appendix C), it is proved that number of iterations of each KM algorithm is $\leq \lfloor (n+1)/2 \rfloor$, which is much smaller than n. Many simulation studies have been performed in which it has been observed that for two significant figures (often this accuracy is adequate for a fuzzy system) the KM and EKM algorithms achieve their final results in from two to six iterations, *regardless of* n. Most recently, Salaken et al. (2016) have shown that their modified EKM algorithms[10] achieve their final results in from one to two iterations, *regardless of* n. An interesting question is: Can the *rate of convergence* be quantified? This question is studied in Sect. 2.2 of Appendix 2.

8.3 Type-Reduction for IT2 FSs and Fuzzy Systems

Because IT2 FSs are much simpler than GT2 FSs, the focus in this section is on type-reduction for the former, whereas the focus in Sect. 8.4 is on type-reduction for the latter. Interval type-2 fuzzy systems, as covered in Chap. 9, will only use three kinds of type-reduction—centroid, height, and center-of-sets (COS). It will be demonstrated in this section that each of these three kinds of type-reduction can be computed by using an IWA, and this will be done without needing explicit knowledge from Chap. 9.

Regardless of the kind of type-reduction, the following is always true: *type-reduction is an extension of a type-1 defuzzification procedure, obtained by using the Extension Principle, as expressed in* (2.71) *in which the t-norm is either the minimum or the product*. For an IT2 FS, it does not matter which one of these t-norms is used because all of the secondary grades are equal to 1 [and $\min(1,1)=1$, $1 \times 1 = 1$].

[10]The Modified EKM algorithms are (see Table 8.5): (1) EKM algorithm is unchanged for y_l, and (2) Step 1 of the EKM algorithm for y_r is changed to "$k = [n/1.6]$ …".

8.3.1 Centroid Type-Reduction for IT2 Fuzzy Sets

Unlike height and center-of-sets type-reduction, which can only be applied to fuzzy sets that occur within an IT2 rule-based fuzzy system (this is analogous to height and center-of-sets defuzzification that only occur within a T1 fuzzy system), centroid type-reduction can be applied to any IT2 FS. Consequently, in the rest of this section no connection is made for IT2 FS $\tilde{A}$ to an IT2 fuzzy system. This connection is made in Sect. 8.3.2

Definition 8.2 The *centroid* $C_{\tilde{A}}(x)$ of $\tilde{A}$ is the union of the centroids, $c(A_e)$, of all its embedded type-1 fuzzy sets A_e, and associated with each of these numbers is a membership grade of 1, because the secondary grades of an IT2 FS are all equal to 1. This means (Karnik and Mendel 2001; Mendel 2001, Chap. 9):

$$\begin{aligned} C_{\tilde{A}}(x) &= 1\Big/\bigcup_{\forall A_e} c_{\tilde{A}}(A_e) = 1\Big/\bigcup_{\forall A_e} \frac{\sum_{i=1}^{N} x_i \mu_{A_e}(x_i)}{\sum_{i=1}^{N} \mu_{A_e}(x_i)} \\ &= 1/\{c_l(\tilde{A}),\ldots,c_r(\tilde{A})\} \equiv 1/[c_l(\tilde{A}),c_r(\tilde{A})] \end{aligned} \tag{8.21}$$

where N is the number of samples of the support of the UMF of $\tilde{A}$, and

$$c_l(\tilde{A}) = \min_{\forall A_e} c_{\tilde{A}}(A_e) = \min_{\forall w_i \in \left[\underline{\mu}_{\tilde{A}}(x_i),\bar{\mu}_{\tilde{A}}(x_i)\right]} \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{8.22}$$

$$c_r(\tilde{A}) = \max_{\forall A_e} c_{\tilde{A}}(A_e) = \max_{\forall w_i \in \left[\underline{\mu}_{\tilde{A}}(x_i),\bar{\mu}_{\tilde{A}}(x_i)\right]} \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{8.23}$$

Note that (8.21) derives from the Wavy-Slice Representation Theorem 6.3 for IT2 FSs and the Extension Principle, as stated in (2.71), and (8.22) and (8.23) use the equivalent vertical slice representation for $\tilde{A}$.

Theorem 8.1 *$[c_l(\tilde{A}),c_r(\tilde{A})]$ is an IWA in which $y_l = c_l(\tilde{A})$, $y_r = c_r(\tilde{A})$, $n = N$, $a_i = b_i = x_i$, $c_i = \underline{\mu}_{\tilde{A}}(x_i)$ and $d_i = \bar{\mu}_{\tilde{A}}(x_i)$, so that:*

$$c_l(\tilde{A}) = \frac{\sum_{i=1}^{L} x_i \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{L} x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{L} \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{L} \underline{\mu}_{\tilde{A}}(x_i)} \tag{8.24}$$

$$c_r(\tilde{A}) = \frac{\sum_{i=1}^{R} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} x_i \bar{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{R} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} \bar{\mu}_{\tilde{A}}(x_i)} \tag{8.25}$$

Proof Compare (8.22) and (8.23) to (8.5) and (8.6), respectively, and subsequently (8.24) and (8.25) to (8.15) and (8.16), respectively.

Table 8.9 Centroid results for a Gaussian primary MF with uncertain mean (Mendel 2005)

$[m_1, m_2]$	$m_2 - m_1$	$[c_l, c_r]$	$c_r - c_l$	$(c_r + c_l)/2$
[5, 5]	0	[5, 5]	0	5
[4.87, 5.12]	0.25	[4.87, 5.12]	0.25	5
[4.75, 5.25]	0.5	[4.74, 5.25]	0.51	5
[4.62, 5.37]	0.75	[4.62, 5.37]	0.75	5
[4.50, 5.50]	1	[4.49, 5.50]	1.01	5
[4.25, 5.75]	1.5	[4.21, 5.78]	1.57	5
[4, 6]	2	[3.90, 6.09]	2.19	5
[3.75, 6.25]	2.5	[3.55, 6.44]	2.89	5
[3.50, 6.50]	3	[3.15, 6.84]	3.69	5

In (8.21), $C_{\tilde{A}}(x)$ is shown as an explicit function of x because the centroid of each embedded type-1 fuzzy set falls on the x-axis. Note that it is customary in the IT2 FS literature to call $[c_l(\tilde{A}), c_r(\tilde{A})]$ the centroid of $\tilde{A}$, ignoring the uninformative membership function grade of 1, and $C_{\tilde{A}}$ is sometimes used instead of $C_{\tilde{A}}(x)$.

As a result of Theorem 8.1, it is easy to compute $C_{\tilde{A}}(x)$ by algorithms such as EIASC or EKM.

Example 8.6 (Mendel 2005) In this example, the centroid is computed for the Example 6.17 Gaussian primary MF with uncertain mean $m \in [m_1, m_2]$ and certain standard deviation ($\sigma = 1$, see Fig. 6.17). Table 8.9 summarizes the results for a range of $m_2 - m_1$ values, including the type-1 case when $m_2 - m_1 = 0$. Observe that:

- As the uncertainty about the mean increases, $c_r - c_l$ increases.
- $[c_l, c_r]$ is always symmetrical about the type-1 mean $m = 5$, and the average value of c_l and c_r is always equal to 5, regardless of the amount of uncertainty there is in m.
- Karnik and Mendel (1998) have proven that if $m_2 - m_1$ is small compared to the standard deviation (σ) of each Gaussian, then $[c_l, c_r] \approx [m_2, m_1]$. In this example, $\sigma = 1$ and the results in Table 8.9 support this theoretical result.

Example 8.7 (Mendel 2005) In this example, the centroid is computed for the Example 6.16 Gaussian primary MF with uncertain standard deviation $\sigma \in [\sigma_1, \sigma_2]$ and certain mean ($m = 5$, see Fig. 6.16). Table 8.10 summarizes the results for a range of $\sigma_2 - \sigma_1$ values, including the type-1 case when $\sigma_2 - \sigma_1 = 0$. Observe that:

- As the uncertainty about the standard deviation increases, $c_r - c_l$ increases.
- $[c_l, c_r]$ is again always symmetrical about the known mean $m = 5$, and the average value of c_l and c_r is always equal to 5, regardless of the amount of uncertainty there is in σ.
- $[c_l, c_r]$ is not close to $[\sigma_2, \sigma_1]$, even for small values of $\sigma_2 - \sigma_1$.

Table 8.10 Centroid results for a Gaussian primary MF with uncertain standard deviation (Mendel 2005)

$[\sigma_1, \sigma_2]$	$\sigma_2 - \sigma_1$	$[c_l, c_r]$	$c_r - c_l$	$(c_r + c_l)/2$
[1, 1]	0	[5, 5]	0	5
[0.88, 1.13]	0.25	[4.80, 5.20]	0.40	5
[0.75, 1.25]	0.5	[4.60, 5.40]	0.80	5
[0.63, 1.38]	0.75	[4.40, 5.60]	1.20	5
[0.50, 1.50]	1	[4.18, 5.81]	1.62	5
[0.38, 1.63]	1.25	[3.93, 6.07]	2.14	5
[0.25, 1.75]	1.5	[3.59, 6.41]	2.82	5

The second observation in Examples 8.6 and 8.7 has led to the following theoretical results whose proofs[11] are in Mendel (2005):

1. Given the FOU of an IT2 FS, one that is *symmetrical* about the primary variable x at $x = m$, then the centroid of such a type-2 fuzzy set is symmetrical about $x = m$, and the average value (i.e., the defuzzified value) of all the elements in the centroid equals m.

2. Given the FOU of an IT2 FS, one that is *symmetrical* about the primary variable x at $x = m$, $[c_l, c_r]$ can be computed by using an algorithm such as EKM or EIASC to compute c_l, after which c_r can be computed as $c_r = 2m - c_l$, leading to a 50% reduction in centroid computations (Exercise 8.6).

3. If all that is desired is a crisp number after performing operations on IT2 FSs, then for the use of such sets to make a difference to not using them (e.g., to using T1 FSs or just crisp numbers) the operations that are applied to them must lead to an IT2 FS that has a nonsymmetrical FOU.

Interestingly enough, nonsymmetrical FOUs usually occur in an IT2 Mamdani fuzzy system when more than one rule fires, as is demonstrated in Chap. 9. This has already been observed for a type-1 fuzzy system in Fig. 3.9, and the same will be observed in Fig. 9.12 for an IT2 fuzzy system. So, even though the MF for the consequent of each fired rule may be symmetrical, this symmetry is fortunately lost when the fired-rule fuzzy sets are combined.

Example 8.8 Suppose the domain of interest is the interval [0, 10] and that three terms are associated with that interval whose type-1 MFs are depicted in Fig. 8.5a.

---

[11]The proofs are not included in this book because, in practice, fully symmetrical FOUs in a rule-based fuzzy system after inferencing are very rare. Additionally, and surprisingly, the proofs [which rely on (8.21)] are not so simple even though the results are almost "obvious." Interestingly, in a symmetrical FOU there will always be mirror image embedded type-1 fuzzy sets (because the left-half of the FOU is the mirror image of the right-half of the FOU), which is why the first result is true.

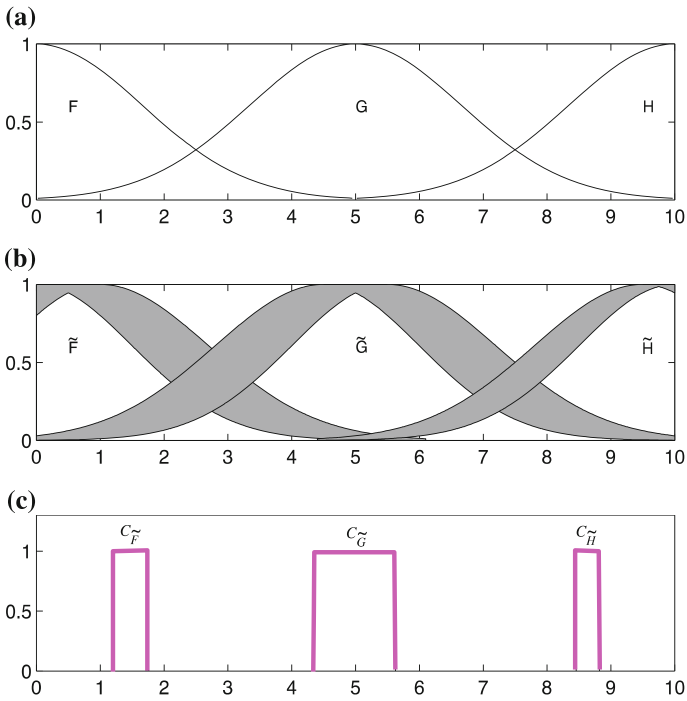


Fig. 8.5 **a** Type-1 MFs, **b** FOUs for their type-2 counterparts, and **c** centroids of the type-2 MFs

Note that the centroids of the three type-1 MFs (F, G, and H) are $c_F = 1.32$, $c_G = 5$, and $c_H = 8.68$. Because of uncertainties about where to center these Gaussian MFs, the FOUs that are depicted in Fig. 8.5b were chosen. The centroids of these IT2 FSs are depicted in Fig. 8.5c. Each centroid was calculated separately for each of the three IT2 FSs using Theorem 8.1 and the EKM algorithms, e.g., $C_{\tilde{F}} = [1.21, 1.76]$. Observe that the midpoint of this centroid is 1.485, which differs from c_F, which demonstrates that *MF uncertainty leads to results that are different than results obtained when MF uncertainty is ignored*. This observation is important when defuzzification is studied for an interval type-2 fuzzy system in Sect. 9.6.

Example 8.9 Figure 8.6 depicts an FOU for an IT2 FS, and four of its embedded T1 FSs. The center of gravity for each of these embedded T1 FSs is (to two significant figures): $c_{(a)} = 4.65$, $c_{(b)} = 4.69$, $c_{(c)} = 3.99$, and $c_{(d)} = 5.37$. By using Theorem 8.1, it is established that $c_l = c_{(c)}$ and $c_r = c_{(d)}$, so that $C_{\tilde{A}}(x) = [3.99, 5.37]$; hence, this example should dispel any mistaken belief that the

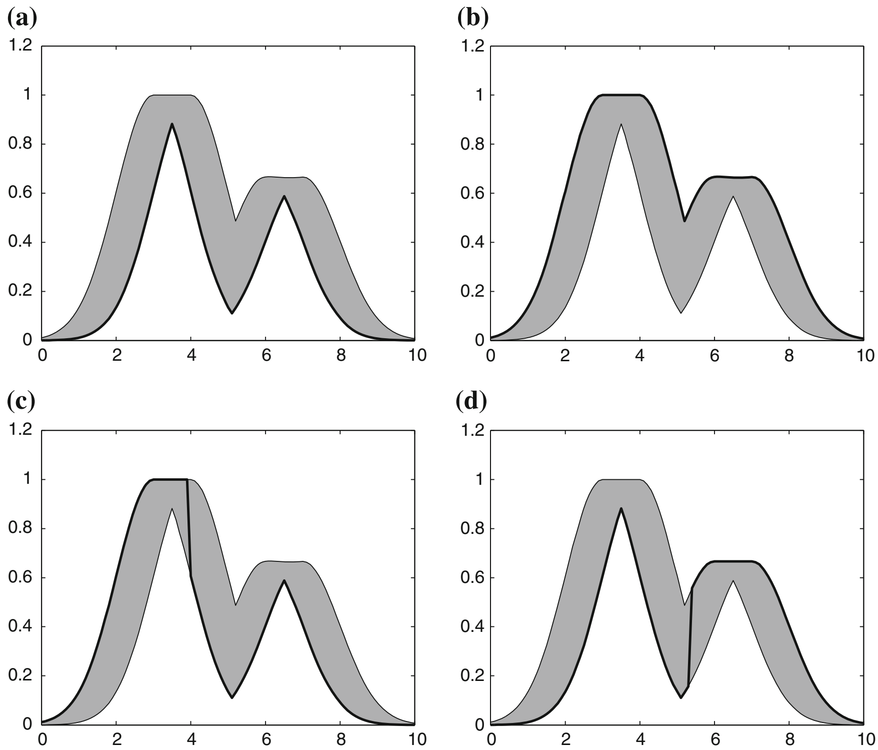


Fig. 8.6 FOU and four embedded T1 FSs. **a** and **b** depict the lower and upper MFs used to compute $c_{(a)}$ and $c_{(b)}$, respectively; **c** and **d** (which are a result of using Theorem 8.1 and EKM algorithms) are associated with c_l and c_r, respectively

end-points of the centroid of an IT2 FS are associated with the centroids of its lower and upper MFs, $c_{(a)}$ and $c_{(b)}$. They are associated with embedded T1 FSs that involve segments from both the lower and upper MFs.

When the black vertical lines in Fig. 8.6c, d are projected down to the x-axis, the reader will observe that they intersect that axis just about at $x = 4$ and $x = 5.4$, respectively. Surprisingly, it is no coincidence that $x = 4 \approx c_l(\tilde{A})$ and $x = 5.4 \approx c_r(\tilde{A})$; this is explained in Sect. 2.2 of Appendix 2.

Some researchers claim that the embedded T1 FSs that are associated with c_l and c_r have no physical meanings, unlike the LMF and UMF which have physical meanings, which makes them uncomfortable. The mathematics which led to the embedded T1 FSs that are associated with c_l and c_r has nothing to do with "physical meaningfulness." To impose "physical meaningfulness" on the optimization problems for c_l and c_r would be to change the present unconstrained optimization problems to constrained optimization problems. To do this, one must first be able to quantify what "physically meaningful embedded T1 FSs" are, which is an open research problem as of the year 2017.

The following are important properties about the centroid of an IT2 FS:

Property 8.1 $C_{\tilde{A}}(x)$ *provides an* uncertainty measure *for IT2 FS* $\tilde{A}$.

It is well known from information theory that entropy provides a measure of the uncertainty of a random variable (Cover and Thomas 1991). Recall that a one-dimensional random variable that is uniformly distributed over a region has entropy equal to the logarithm of the *length* of that region. Comparing the MF $\mu_{C_{\tilde{A}}}(x)$ of $C_{\tilde{A}}(x)$ (where $\mu_{C_{\tilde{A}}}(x) = 1$ when $x \in [c_l, c_r]$ and $\mu_{C_{\tilde{A}}}(x) = 0$ when $x \notin [c_l, c_r]$) with a probability density function $p_X(x)$ that is uniformly distributed over $[c_l, c_r]$ (where $p_X(x) = (c_r - c_l)^{-1}$ when $x \in [c_l, c_r]$ and $p_X(x) = 0$ when $x \notin [c_l, c_r]$), it is clear that they are similar to within a scale factor. It is therefore reasonable to consider the support of $C_{\tilde{A}}(x)$, $c_r - c_l$, as a measure of the extent of the uncertainty of an IT2 FS (Wu and Mendel 2002).

See Examples 8.6 and 8.7 for illustrations that support the use of $C_{\tilde{A}}(x)$ as an uncertainty measure for their respective IT2 FSs.

Property 8.2 (Mendel and Wu 2007) *Let* $\tilde{A}$ *be an IT2 FS defined on X, and* $\tilde{A}'$ *be* $\tilde{A}$ *shifted by*[12] Δm *along X, i.e,.* $\underline{\mu}_{\tilde{A}'}(x) = \underline{\mu}_{\tilde{A}}(x - \Delta m)$ *and* $\bar{\mu}_{\tilde{A}'}(x) = \bar{\mu}_{\tilde{A}'}(x - \Delta m)$. *Then the centroid of* $\tilde{A}'$, $C_{\tilde{A}'}(x) = [c_l(\tilde{A}'), c_r(\tilde{A}')]$, *is the same as the centroid of* $\tilde{A}$, $C_{\tilde{A}}(x) = [c_l(\tilde{A}), c_r(\tilde{A})]$, *shifted by* Δm, *i.e,.* $c_l(\tilde{A}') = c_l(\tilde{A}) + \Delta m$ *and* $c_r(\tilde{A}') = c_r(\tilde{A}) + \Delta m$.

This property lets $\mathrm{FOU}(\tilde{A})$ be relocated to a more convenient place for the actual computations of c_l and c_r, and demonstrates that it is only the shape of $\mathrm{FOU}(\tilde{A})$ that affects $c_r - c_l$ and not where that shape resides on the axis of the primary variable. A proof of this property for a continuous universe of discourse is left as an exercise for the reader (Exercise 8.7)

Property 8.3 (Mendel and Wu 2007) *If the primary variable x is bounded, i.e.,* $x \in [x_1, x_N]$, *then* $c_l(\tilde{A}) \geq x_1$ *and* $c_r(\tilde{A}) \leq x_N$.

Although the centroid cannot be computed in closed form, this property provides bounds for the centroid that are available from knowledge of the domain of the primary variable. Its proof is also left as an exercise for the reader (Exercise 8.8).

Property 8.4 (Mendel and Wu 2007) *If* $\mathrm{LMF}(\tilde{A})$ *is entirely on the primary variable (x) axis, and* $x \in [x_1, x_N]$, *then the centroid does not depend upon the shape of* $\mathrm{FOU}(\tilde{A})$ *and, as long as* x_1 *and* x_N *are included in the sampling points, the centroid equals* $[x_1, x_N]$.

[12] Δm can be positive or negative.

An example of an FOU for which LMF($\tilde{A}$) is entirely on the primary variable (x) axis, and $x \in [x_1, x_N]$, is depicted in Fig. 8.7. It is called a *completely filled-in FOU*. While the results of this property may seem strange, remember that each of the centroids in (8.21) that make up the centroid of $\tilde{A}$ provides a center of gravity about the vertical (primary membership) axis, and according to Property 8.3 each of these centroids must be contained within $[x_1, x_N]$. For a completely filled-in FOU, the centroid actually equals $[x_1, x_N]$. This property also demonstrates that for such an FOU the functional form of its UMF plays no role in determining the centroid. In this book, this is considered to be an undesirable property for an FOU.

Proof of Property 8.4 (Mendel and Wu 2007, pp. 375–376) To show $c_l = x_1$ consider the embedded T1 FS (see Fig. 8.8) $\{(x_1, 0), (x_2, \varepsilon), (x_3, 0), \ldots, (x_N, 0)\}$. When $\varepsilon \to 0$ the sampling approaches zero, and $x_2 \to x_1$. It follows that

$$\sum_{i=1}^{N} x_i w_i \Bigg/ \sum_{i=1}^{N} w_i = \frac{x_2 \varepsilon}{\varepsilon} = x_2 \to x_1 \text{ as } \varepsilon \to 0 \tag{8.26}$$

From Property 8.3, $c_l \geq x_1$; hence, this fact together with (8.26) let one conclude that $c_l = x_1$.

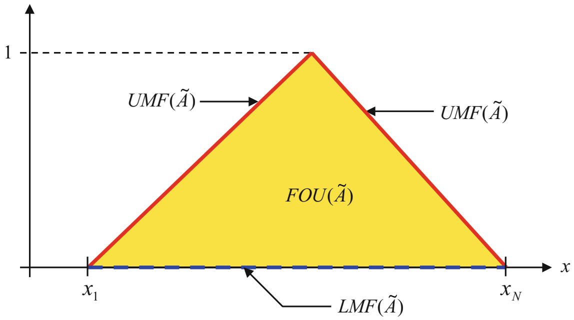


Fig. 8.7 Completely filled-in FOU (Mendel and Wu 2007: © 2007, Elsevier)

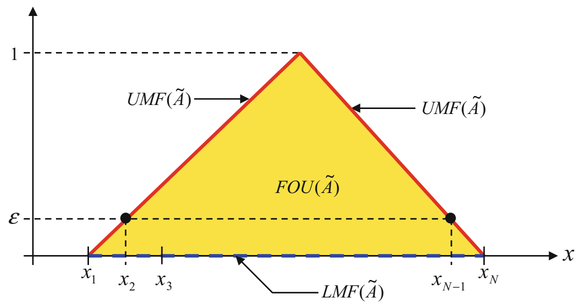


Fig. 8.8 Constructions for the two embedded type-1 fuzzy sets used in the proof of Property 8.4 (Mendel and Wu 2007: © 2007, Elsevier)

To show $c_r = x_N$, consider the embedded T1 FS (see Fig. 8.8) $\{(x_1, 0), (x_2, 0), \ldots, (x_{N-1}, \varepsilon), (x_N, 0)\}$. When $\varepsilon \to 0$ the sampling approaches zero, and $x_{N-1} \to x_N$. It follows that

$$\sum_{i=1}^{N} x_i w_i \Big/ \sum_{i=1}^{N} w_i = \frac{x_{N-1}\varepsilon}{\varepsilon} = x_{N-1} \to x_N \text{ as } \varepsilon \to 0 \tag{8.27}$$

From Property 8.3, $c_r \leq x_N$; hence, this fact together with (8.27) let one conclude that $c_r = x_N$. Q. E. D.

Property 8.5 (Wu and Nie 2011) *When $\tilde{A} = \tilde{A}(t)$, $t = t_1, t_2, \ldots$, and $\tilde{A}(t_{k+1})$ does not change much from $\tilde{A}(t_k)$, then the EKM algorithms for computing $C_{\tilde{A}(t_{k+1})}(x)$ should be initialized by using $c_l(\tilde{A}(t_k))$ and $c_r(\tilde{A}(t_k))$. Doing this is called "EKMANI" (Enhanced KM algorithm with new initialization) and a fewer number of iterations will be required of the EKM algorithms when it is used then if it is not used.*

No proof is needed of this property, because it is self-evident. It provides a very practical way to initialize the EKM algorithms and *should be adopted for all real-time IT2 fuzzy systems*, e.g., fuzzy logic control, time-series forecasting, etc.

8.3.2 Centroid Type-Reduction in an IT2 Fuzzy System[13]

Recall (Chap. 3) that when an input $\mathbf{x} = \mathbf{x}'$ is applied to a type-1 rule it leads to a *firing level* for the rule, $f^l(\mathbf{x}')$, which may then be combined, in a Mamdani fuzzy system, with the entire consequent of the rule, G^l, by means of a t- norm, leading to a *type-1 fired-rule output fuzzy set*, B^l, after which all of the fired type-1 rule output fuzzy sets may be combined by means of the union operation, producing one *combined type-1 fired-rule output fuzzy set*, B, which is then *defuzzified* by computing its centroid to give $y_c(\mathbf{x}')$.

Chapter 9 will demonstrate that, when an input $\mathbf{x} = \mathbf{x}'$ is applied to an IT2 rule (i.e., a rule that looks just like a type-1 rule except that some or all of its fuzzy sets are IT2 FSs) it leads to a *firing interval* $[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]$ which may then be combined with the entire consequent of that rule, $\tilde{G}^l$, by means of the meet operation, leading to an IT2 *fired-rule output FS*, $\tilde{B}^l$. Then, all of the IT2 fired-rule output FSs may be combined by means of the join operation, producing one *combined IT2 fired-rule output fuzzy set*, $\tilde{B}$. $\tilde{B}$ is then *type-reduced* by computing its centroid to give $C_{\tilde{B}}$, which is computed as explained in Sect. 8.3.1. So, Theorem 8.1 applies directly to centroid type-reduction in a type-2 fuzzy system.

---

[13]Some readers may prefer to read this section after reading Sect. 9.6.1.

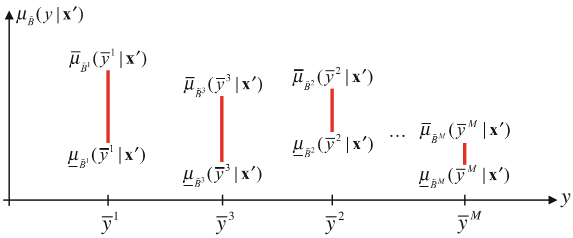


Fig. 8.9 Intervals $[\underline{\mu}_{\tilde{B}^l}(\bar{y}^l|\mathbf{x}'), \bar{\mu}_{\tilde{B}^l}(\bar{y}^l|\mathbf{x}')]$ located at $\bar{y}^l(l = 1, \ldots, M)$. Intervals do not have to appear in chronological order

8.3.3 *Height Type-Reduction in an IT2 Fuzzy System*

Recall that for a type-1 fuzzy system, the height defuzzifier (Sect. 3.6.2) replaces the fired-rule output set of each fired rule, B^l, by a singleton at the point having maximum membership in the rule's consequent fuzzy set, with amplitude equal to the fired-rule output's MF at that point, and then calculates the centroid of the type-1 set comprised of these singletons (see Fig. 3.11). The output of a height defuzzifier, $y_h(\mathbf{x}')$, is given as (3.39)

$$y_h(\mathbf{x}') = \frac{\sum_{l=1}^{M} \bar{y}^l \mu_{B^l}(\bar{y}^l|\mathbf{x}')}{\sum_{l=1}^{M} \mu_{B^l}(\bar{y}^l|\mathbf{x}')} \tag{8.28}$$

In (8.28), $\bar{y}^l$ is the point having maximum membership in the lth consequent fuzzy set (if there is more than one such point, their average can be taken as $\bar{y}^l$), and its membership grade in the lth fired—rule output set is $\mu_{B^l}(\bar{y}^l|\mathbf{x}')$.

The *height type-reducer* replaces each fired-rule IT2 output fuzzy set $\tilde{B}^l$ by an IT2 FS whose y-domain consists of a single point ($\bar{y}^l$), and whose secondary MF is the type-1 interval fuzzy number (Definition 2.6) $\mu_{\tilde{B}^l}(\bar{y}^l|\mathbf{x}') = 1/[\underline{\mu}_{\tilde{B}^l}(\bar{y}^l|\mathbf{x}'), \bar{\mu}_{\tilde{B}^l}(\bar{y}^l|\mathbf{x}')]$. $\bar{y}^l$ can be chosen to be the point having the highest primary membership in the principal MF[14] (Definition 6.13) of the output set $\tilde{B}^l$ (e.g., in Figs. 6.16 and 6.17, $\bar{y}^l$ would be set equal to 5). Formulas for $\underline{\mu}_{\tilde{B}^l}(\bar{y}^l|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}^l}(\bar{y}^l|\mathbf{x}')$ are given in Chap. 9. This procedure is summarized in Fig. 8.9, which is the IT2 version of Fig. 3.11b.

Let $Y_h(\mathbf{x}')$ denote the height type-reduced set, where

$$Y_h(\mathbf{x}') = 1 \Bigg/ \frac{\sum_{i=1}^{M} \bar{y}^i \mu_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')}{\sum_{i=1}^{M} \mu_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')} \tag{8.29}$$

[14] If $\tilde{B}^l$ is such that a principal MF cannot be defined, one may choose $\bar{y}^l$ as the point having the highest primary membership with a secondary grade equal to 1, or as a point satisfying some similar criterion.

in which $(i = 1, \ldots, M)$

$$\mu_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') \in [\underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}'), \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')] \tag{8.30}$$

Each of the intervals $[\underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}'), \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')]$ can be discretized, after which, in (8.29), $\sum_{i=1}^{M} \bar{y}^i \mu_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') / \sum_{i=1}^{M} \mu_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')$ can be computed a multitude of times, each time using one discrete point from each of these intervals. Clearly, doing this will lead to an interval of real numbers, $[y_l^h(\mathbf{x}'), y_r^h(\mathbf{x}')]$, where:

$$y_l^h(\mathbf{x}') = \min_{w_i \in [\underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}'), \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')]} \frac{\sum_{i=1}^{M} \bar{y}^i w_i}{\sum_{i=1}^{M} w_i} \tag{8.31}$$

$$y_r^h(\mathbf{x}') = \max_{w_i \in [\underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}'), \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')]} \frac{\sum_{i=1}^{M} \bar{y}^i w_i}{\sum_{i=1}^{M} w_i} \tag{8.32}$$

so that

$$Y_h(\mathbf{x}') = 1/[y_l^h(\mathbf{x}'), y_r^h(\mathbf{x}')] \tag{8.33}$$

Theorem 8.2 *$[y_l^h(\mathbf{x}'), y_r^h(\mathbf{x}')]$ is an IWA in which $y_l = y_l^h(\mathbf{x}')$, $y_r = y_r^h(\mathbf{x}')$, $n = M$, $a_i = b_i = \bar{y}^i$, $c_i = \underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')$ and $d_i = \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')$, so that*

$$y_l^h(\mathbf{x}') = \frac{\sum_{i=1}^{L} \bar{y}^i \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') + \sum_{i=L+1}^{M} \bar{y}^i \underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')}{\sum_{i=1}^{L} \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') + \sum_{i=L+1}^{M} \underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')} \tag{8.34}$$

$$y_r^h(\mathbf{x}') = \frac{\sum_{i=1}^{R} \bar{y}^i \underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') + \sum_{i=R+1}^{M} \bar{y}^i \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')}{\sum_{i=1}^{R} \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') + \sum_{i=R+1}^{M} \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')} \tag{8.35}$$

Proof Compare (8.31) and (8.32) to (8.5) and (8.6), respectively, and subsequently (8.34) and (8.35) to (8.15) and (8.16), respectively.

An example in which $Y_h(\mathbf{x}')$ is computed, is given in Sect. 8.3.5.

An important observation about height type-reduction is that it does not begin with an FOU or DOU. Even though one may interpret the red vertical lines in Fig. 8.9 as describing a DOU, this DOU was created by means of a construction, and since there is no natural order for rules, different looking DOUs could have been obtained simply by renumbering the M rules. Doing this would not change $y_l^h(\mathbf{x}')$ and $y_r^h(\mathbf{x}')$ (Exercise 8.10).

8.3.4 Center-of-Sets (COS) Type-Reduction in an IT2 Fuzzy System

Recall that for a type-1 fuzzy system, the COS defuzzifier (Sect. 3.6.3) replaces each rule consequent T1 FS, G^l, by a singleton located at its centroid, c^l, with an amplitude equal to the firing level, $f^l(\mathbf{x}')$, after which the centroid of these singletons is found. The output of a COS defuzzifier, $y_{\text{COS}}(\mathbf{x}')$, is given, as (3.41):

$$y_{\text{COS}}(\mathbf{x}') = \frac{\sum_{l=1}^{M} c^l f^l(\mathbf{x}')}{\sum_{l=1}^{M} f^l(\mathbf{x}')} \tag{8.36}$$

The *COS type-reducer* replaces each rule consequent IT2 FS, $\tilde{G}^i$, by the support of its centroid, $[c_l(\tilde{G}^i), c_r(\tilde{G}^i)]$, and assigns a secondary MF of $1/[\underline{f}^i(\mathbf{x}'), \bar{f}^i(\mathbf{x}')]$ to it, where $[\underline{f}^i(\mathbf{x}'), \bar{f}^i(\mathbf{x}')]$ is the firing interval for the ith rule. Formulas for $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are given in Chap. 9. This procedure is summarized in Fig. 8.10 [in which $c_l(\tilde{G}^i)$ is shortened to c_l^i and $c_r(\tilde{G}^i)$ is shortened to c_r^i], which is the IT2 version of Fig. 3.12. Observe that the shaded rectangles in Fig. 8.10 are information granules (Definition 6.16), and, although not shown on Fig. 8.10, these granules may overlap. If this happens then each granule is still treated as a separate entity for COS type-reduction.

Let $Y_{\text{COS}}(\mathbf{x}')$ denote the COS type-reduced set, where

$$Y_{\text{COS}}(\mathbf{x}') = 1 \Bigg/ \frac{\sum_{i=1}^{M} y_i w_i}{\sum_{i=1}^{M} w_i} \tag{8.37}$$

in which $(i = 1, \ldots, M)$

$$y_i \in [c_l(\tilde{G}^i), c_r(\tilde{G}^i)] \tag{8.38}$$

$$w_i \in [\underline{f}^i(\mathbf{x}'), \bar{f}^i(\mathbf{x}')] \tag{8.39}$$

Each of the firing intervals $[\underline{f}^i(\mathbf{x}'), \bar{f}^i(\mathbf{x}')]$ and $[c_l(\tilde{G}^i), c_r(\tilde{G}^i)]$ can be discretized, after which, in (8.37), $\sum_{i=1}^{M} y_i w_i / \sum_{i=1}^{M} w_i$ can be computed a multitude of times,

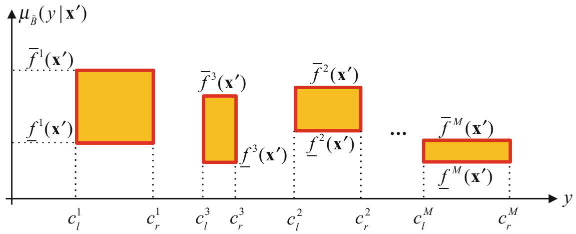


Fig. 8.10 Granules located at $[c_l^i, c_r^i]$. Granules do not have to appear in chronological order

each time using one discrete point from each of these intervals. Clearly, doing this will again lead to an interval of real numbers $[y_l^{\mathrm{COS}}(\mathbf{x}'), y_r^{\mathrm{COS}}(\mathbf{x}')]$, where:

$$y_l^{\mathrm{COS}}(\mathbf{x}') = \min_{w_i \in [\underline{f}^i(\mathbf{x}'), \bar{f}^i(\mathbf{x}')],\, y_i \in [c_l(\tilde{G}^i), c_r(\tilde{G}^i)]} \frac{\sum_{i=1}^{M} y_i w_i}{\sum_{i=1}^{M} w_i} \tag{8.40}$$

$$y_r^{\mathrm{COS}}(\mathbf{x}') = \max_{w_i \in [\underline{f}^i(\mathbf{x}'), \bar{f}^i(\mathbf{x}')],\, y_i \in [c_l(\tilde{G}^i), c_r(\tilde{G}^i)]} \frac{\sum_{i=1}^{M} y_i w_i}{\sum_{i=1}^{M} w_i} \tag{8.41}$$

so that

$$Y_{\mathrm{COS}}(\mathbf{x}') = 1/[y_l^{\mathrm{COS}}(\mathbf{x}'), y_r^{\mathrm{COS}}(\mathbf{x}')] \tag{8.42}$$

Theorem 8.3 $[y_l^{\mathrm{COS}}(\mathbf{x}'), y_r^{\mathrm{COS}}(\mathbf{x}')]$ *is an* IWA *in which* $y_l = y_l^{\mathrm{COS}}(\mathbf{x}')$, $y_r = y_r^{\mathrm{COS}}(\mathbf{x}')$, $n = M$, $a_i = c_l(\tilde{G}^i)$, $b_i = c_r(\tilde{G}^i)$, $c_i = \underline{f}^i(\mathbf{x}')$ *and* $d_i = \bar{f}^i(\mathbf{x}')$, *so that*

$$y_l^{\mathrm{COS}}(\mathbf{x}') = \frac{\sum_{i=1}^{L} c_l(\tilde{G}^i)\bar{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M} c_l(\tilde{G}^i)\underline{f}^i(\mathbf{x}')}{\sum_{i=1}^{L} \bar{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M} \underline{f}^i(\mathbf{x}')} \tag{8.43}$$

$$y_r^{\mathrm{COS}}(\mathbf{x}') = \frac{\sum_{i=1}^{R} c_r(\tilde{G}^i)\underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M} c_r(\tilde{G}^i)\bar{f}^i(\mathbf{x}')}{\sum_{i=1}^{R} \underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M} \bar{f}^i(\mathbf{x}')} \tag{8.44}$$

Proof Compare (8.40) and (8.41) to (8.5) and (8.6), respectively, and subsequently (8.43) and (8.44) to (8.15) and (8.16), respectively.

Observe that to compute $Y_{\mathrm{COS}}(\mathbf{x}')$ one must first compute the centroid of each rule's IT2 consequent set, $[c_l(\tilde{G}^i), c_r(\tilde{G}^i)]$, which can be done by using Theorem 8.1, and that this only has to be done once after an IT2 fuzzy system has been designed, because those centroids do not depend upon the input $\mathbf{x} = \mathbf{x}'$ to the fuzzy system.

Observe, also, that, just as for height type-reduction, COS type-reduction does not begin with an FOU or DOU. Even though one may interpret the filled-in granules in Fig. 8.10 as describing a DOU, this DOU was created by means of a construction, and since there is no natural order for rules, different looking DOUs could have been obtained simply by renumbering the M rules. Doing this would not change $y_l^{\mathrm{COS}}(\mathbf{x}')$ and $y_r^{\mathrm{COS}}(\mathbf{x}')$ (Exercise 8.11).

8.3.5 Type-Reduction Example

This example (Karnik et al. 1999) illustrates the use of the just described three type-reduction methods for an IT2 fuzzy system. To keep things very simple at this point in the book, only a single-input–single-output IT2 fuzzy system is considered, one that has rules of the form $(l = 1, 2, 3)$:

$$\tilde{R}_Z^l : \text{ IF } x \text{ is } \tilde{F}^l, \quad \text{THEN } y \text{ is } \tilde{G}^l \tag{8.45}$$

where $x, y \in [0, 10]$.

Figure 8.11a, b depict the antecedent and consequent IT2 FS FOUs; each FOU is described by Gaussian lower and upper MFs that have the same mean (m) and standard deviation (σ), but the two Gaussians are scaled to different heights. The maximum height reached by the UMF is unity, whereas that reached by the LMF is s. The support of the secondary MF when $x = x'$, $I_{x'}$, for the antecedent FOUs is the interval

$$\left[s_i \exp\left\{ -0.5\left(\frac{x' - m_i}{\sigma_i}\right)^2 \right\}, \exp\left\{ -0.5\left(\frac{x' - m_i}{\sigma_i}\right)^2 \right\} \right].$$

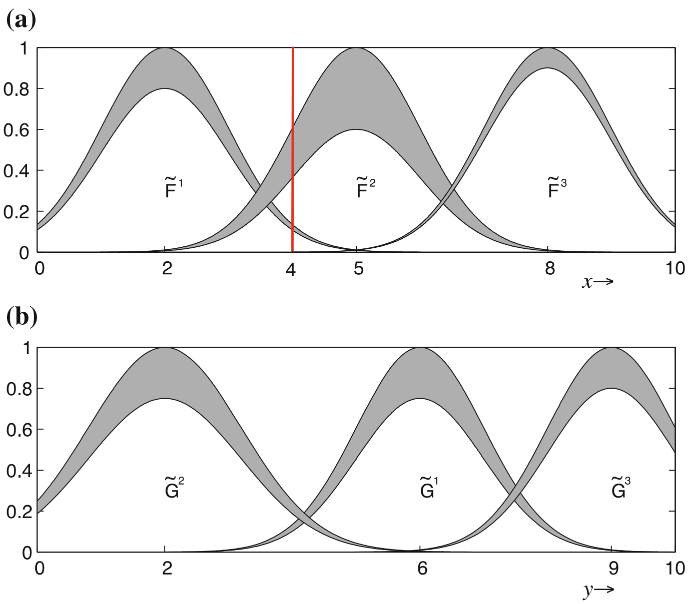


Fig. 8.11 **a** Antecedent set FOUs (the vertical axis shows the primary memberships of x in the antecedent sets) and **b** consequent set FOUs (the vertical axis shows the primary memberships of y in the consequent sets). The applied input (x = 4) is shown in (**a**) (Karnik et al. 1999, © 1999, IEEE)

Table 8.11 Type-reduced results (Karnik et al. 1999)

Type-reduced set	Interval	Center	Spread
Centroid	$[2.33, 3.31]$	2.82	0.49
Height	$[2.47, 3.33]$	2.90	0.43
Center-of-sets	$[2.58, 3.32]$	2.95	0.37

A similar interval applies for the consequent FOUs.

The m_i values for each of the antecedent sets, ($\tilde{F}^1$, $\tilde{F}^2$, and $\tilde{F}^3$), are 2, 5, and 8, respectively; their σ_i values are all the same and are equal to 1; and, their s_i values are 0.8, 0.6, and 0.9, respectively. For the three consequent sets ($\tilde{G}^1$, $\tilde{G}^2$, and $\tilde{G}^3$), their m_i values are 6, 2, and 9, respectively; their σ_i values are 1, 1.2, and 1, respectively; and, their s_i values are 0.75, 0.75, and 0.8, respectively.

The applied input is $x' = 4$, as shown by the red vertical line in Fig. 8.11a. It has nonzero memberships in two antecedent sets, $\tilde{F}_1$ and $\tilde{F}_2$, and so two rules are activated: IF x is $\tilde{F}^1$, THEN y is $\tilde{G}^1$ and IF x is $\tilde{F}^2$, THEN y is $\tilde{G}^2$.

The type-reduced results for this example,[15] which were obtained as described in Sects. 8.3.2–8.3.4, are collected in Table 8.11 in which the domain of the type-reduced set is expressed both as an interval and in terms of its center and spread. Observe that there are some differences between the three type-reduced sets. Those differences are due mainly to the truncated natures of $\tilde{F}^1$ and $\tilde{G}^2$.

8.3.6 *Remarks and Insights*

In this section, some discussions are provided about the history of type-reduction, other type-reduction algorithms, computation time as a metric and accuracy as a metric.

8.3.6.1 History of Type-Reduction

Type-reduction originated in Karnik and Mendel (1998, 2001) out of the necessity to figure out a way to go from a T2 FS to a number. Their approach [as is also given in the first edition of this book (Mendel 2001, Chap. 9)] was different from the one that has been used herein to define the centroid of an IT2 FS and the three kinds of type-reduction that may be used by a type-2 fuzzy system. It began by defining the centroid of a GT2 FS (and not the centroid of an IT2 FS) by using the Extension Principle and the Wavy-Slice Representation Theorem (although it was not given that name nor any other name), and by then computing the centroid for each

[15] Chapter 9 formulas are needed for the firing interval and fired-rule output set in order to compute these type-reduced sets; they are asked for in Exercise 9.30.

embedded T2 FS. Numerical procedures were then stated that required the exhaustive enumeration of all such sets. For height and center-of-sets type-reduction a "generalized centroid" was introduced, and then numerical procedures were again stated that required the exhaustive enumeration of all embedded T2 FSs. Appendix 1 summarizes this earlier approach.

It was already known in Karnik and Mendel (1998, 2001) and Mendel (2001) that exhaustive enumeration was unnecessary for IT2 FSs. In fact, after mentioning the high computational complexity for computing the centroid or generalized centroid for GT2 FSs by exhaustive enumeration procedures, Mendel (2001) focused on IT2 FSs, stating:

> When the secondaries are interval sets [type-1 interval fuzzy numbers], then *exact* results for the centroid or generalized centroid can be determined using a totally different computational procedure, as we explain in the next section. The computational complexity in this case is so low that the centroid and generalized centroid calculations become very practical.

In that "next section" the KM algorithms were derived and then applied to compute the centroid of an IT2 FS and to compute centroid, height and center-of-sets type-reduced sets for a type-2 fuzzy system.

It was not until (Liu 2008) that exhaustive enumeration no longer was necessary for type-reduction of a GT2 FS. The horizontal-slice representation of a GT2 FS made this possible.

8.3.6.2 Other Type-Reduction Algorithms

For readers who want to learn a lot about other algorithms for type-reduction, see Wu (2013) and Mendel (2013). Wu's section on "Enhancements to the KM TR algorithms" and Mendel's sections on "Improved KM algorithms," "Understanding the KM/EKM algorithms, leading to further improved algorithms," and "Eliminating the need for the KM algorithms: Non-KM algorithms/methods that preserve the ability to approximate the centroid or type-reduced set," contain the algorithms or information about them.

The following are papers with other algorithms or modified algorithms that lead to a centroid type-reduced set (given in chronological order): Niewiadomski et al. (2006), Melgarejo (2007), Duran et al. (2008), Li et al. (2008), Starczewski (2009), Liu and Mendel (2011), Hu et al. (2012a, b), Liu et al. (2012a, b), Wu et al. (2012), Ulu et al. (2013) and Chen et al. (2014).

The KM, EKM, and EIASC algorithms are arguably the simplest to derive and explain, which is why only they are given in this chapter. Some of the other algorithms require either that $a_i = b_i \equiv x_i$ and are therefore less general than the KM, EKM, and EIASC algorithms, or have derivations that are very complicated, or are restricted in other ways (e.g., to certain kinds of MFs or FOUs).

The many algorithms for computing y_l and y_r raise the following question: Why is it that so many people have and continue to search for better algorithms for computing y_l and y_r? Having thought about this a lot, my answer is: y_l and y_r are associated with very well defined and relatively easy optimization problems that lend themselves to analyses and/or simulations (e.g., see Appendix 2), and researchers who look at such problems love to do analysis and/or simulations (myself included).

8.3.6.3 Computation Time as a Metric

Many papers about new or improved algorithms (including this author's) provide separate simulation results for both the number of iterations required for their convergence and overall computation time. The former does not change as computers or hardware change or improve, but the latter does; hence, a more meaningful metric would be *computation time per iteration*. This number will, of course, become smaller and smaller as computers become faster and faster, something that always seems to occur. One may conjecture that, at some not-to-distant future time, computation time per iteration will be so small that it will not matter which new or improved algorithm is used, because the differences in overall computation time will be imperceptible to a human. Although this is true for a human, it is very important to realize that when the type-reduction algorithms are implemented in *hardware*, then the faster they can be performed frees up the hardware to perform other computations, something that can be very important for real-world applications. Consequently, it is important to perform the type-reduction calculations as quickly as possible, which is why there has been extensive work on improving type-reduction algorithms. And so, evaluating new type-reduction algorithms in terms of computing time/iteration is a meaningful metric, even when this is done outside of the context of an application of a fuzzy system.

8.3.6.4 Accuracy as a Metric

When only sampled values are given for LMF($\tilde{A}$) and UMF($\tilde{A}$), then the KM, EKM, and EIASC algorithms (and the other algorithms that are referenced in Sect. 8.3.6.2) give *exact* results (see, also the quote in the second paragraph of Sect. 8.3.6.1 in which "exact" is italicized). Of course, when the sampling rate is changed, so that the sampled values change, then different numerical centroid or type-reduced sets will be obtained, but all of these are still *exact* results.

If, on the other hand, one begins with formulas for LMF($\tilde{A}$) and UMF($\tilde{A}$), and wishes to use them to compute the centroid of $\tilde{A}$, then the continuous EKM (CEKM) algorithms that are given in Sect. 2.2 of Appendix 2 can be used. Those algorithms require numerical integrations, which again require some sort of sampling of LMF($\tilde{A}$) and UMF($\tilde{A}$), unless the integrals can be worked out by hand, in which case infinite precision is possible. Approaching infinite precision is also possible by using,[16], e.g., EIASC or EKM algorithms and very fine sampling of $\mathrm{LMF}(\tilde{A})$ and $\mathrm{UMF}(\tilde{A})$.

In the opinion of this author, studies that focus on "accuracy" as a metric for a centroid algorithm, and that evaluate accuracy by using the "exact" centroid, where the "exact" centroid is the infinite precision result, are of limited value, because if centroid type-reduction is used in a type-2 fuzzy system, it is not this kind of "accuracy" that is important. Instead, it is achieving acceptable application-related performance metrics that is important. For a more critical discussion about this, see Mendel (2015, Sect. 3).

Although some people still insist on obtaining the so-called "exact results" by enumerating a very large number of embedded sets, this is unnecessary, and was and is probably due to the appearance of the procedures in Mendel (2001) that required this. It is also probably due to the fact that the IWA was unknown when Mendel (2001) was written, and it took some time for the connections between it and centroid and type-reduction computations to become clear. Liu and Mendel (2008) was the first article where such connections were made; however, because its title does not include the words "centroid" or "type-reduction" and instead uses[17] "fuzzy weighted average," this article arguably went unnoticed by the type-2 community. Further connections between the IWA, the centroid and type-reduction methods appear in Mendel and Wu (2010, Chap. 6), Mendel (2013, Sect. VII.A), Mendel (2014, Sects. VI.A, B), and Mendel et al. (2014, Chap. 3).

8.4 Type-Reduction for GT2 FSs and Fuzzy Systems[18]

This topic has been made more complicated because of the different representations that are now available for GT2 FSs—vertical slice, wavy slice, and horizontal slice. When Mendel (2001) was written, only the vertical-slice representation was known, although the following footnote 2 in its Chap. 7 acknowledged the existence of the wavy-slice representation:

[16]In theory, infinite precision is also possible by using the embedded sets approaches that are described in Appendix 1, and an extremely large number of such sets. The latter makes this approach impractical.

[17]A *fuzzy weighted average* (FWA) is a weighted average in which at least one w_i in (8.1) is modeled as a T1 FS, and the remaining w_i are modeled either as intervals or crisp numbers. Exercise 8.15 is about the FWA and its connections to the IWA. There is even a *linguistic weighted average* (LWA) (Wu and Mendel 2007a, b; Mendel and Wu 2010, Chap. 5) in which at least one w_i in (8.1) is modeled as an IT2 FS, and the remaining w_i are modeled either as T1 FSs, intervals or crisp numbers. LWAs are not needed in this book and so they are not covered herein.

[18]Some readers may prefer to read this section after reading Sect. 11.6.

> Our derivations of union, intersection, and complement of type-2 fuzzy sets have relied on the Extension Principle, because that is what was used by Zadeh and Mizumoto and Tanaka. The Extension Principle, however, is somewhat ad hoc, and deriving things using it may therefore be considered problematic. Very recently, while working with Dr. Robert John, we have been able to derive exactly the same formulas for the union, intersection, and complement of type-2 fuzzy sets without having to use the Extension Principle. Our derivation is based on a new decomposition of a type-2 fuzzy set as a union of all embedded type-2 fuzzy sets [the wavy-slice representation]. Because this work has not yet appeared in any published form, its details are not included here.[19]

Surprisingly, it was the wavy-slice representation (although it was not called this or anything else), and not the vertical-slice representation that was used by Karnik and Mendel (2001) when they proposed the centroid and different kinds of type-reduction for a GT2 FS, and it is their approach that appears in Mendel (2001, Chap. 9).

The dilemma faced by this author in this section is how to choose which representation to use for type-reduction of GT2 FSs. Using the wavy-slice representation, and then providing impractical computing procedures that require the enumeration of embedded T2 FSs no longer makes sense; hence, the approach taken in Mendel (2001, Chap. 9) is no longer emphasized. However, for completeness, and historical reasons, it is described in Appendix 1.

Vertical-slice set theoretic computations for GT2 FSs are very complicated (see Sect. 7.2), since they involve the meet and join operations, and they do not connect to what has already been established in Sect. 8.3 for IT2 FSs; hence, this approach to type-reduction of GT2 FSs is not taken.[20]

People are now performing set theoretic computations for GT2 FSs using horizontal-slice computations (see Sect. 7.4), because each α-plane raised to level α is an IT2 FS whose membership grade is α, so that IT2 FS computations can then be used for GT2 FS computations. Consequently, *the approach to type-reduction that is taken in this section uses the horizontal-slice representation for a GT2 FS*, namely (Sect. 6.7.3):

$$\tilde{A} = \bigcup_{\alpha \in [0,1]} \alpha/\tilde{A}_\alpha = \sup_{\alpha \in [0,1]} \alpha/\tilde{A}_\alpha \tag{8.46}$$

[19] It was published, and is Mendel and John (2002).

[20] A notable exception to the complexity of vertical-slice computations and type-reduction is the *vertical-slice centroid type-reducer* (VSCTR) proposed in John (2000) and then detailed in Lucas et al. (2007). According to Almaraashi et al. (2016, p. 27): "The VSCTR … does not calculate the union [of the centroids] for all the embedded sets involved in the general type-2 fuzzy sets. Although this method does not depend on the concept of embedded sets, it is a good approach for practical usage. This method works as follows: (1) For each vertical slice, the centroid of each vertical slice is calculated exactly as [a] type-1 centroid calculation, and (2) The type-reduced set domain is the same as the vertical slices values; the membership grades of the type-reduced set are the centroids of these vertical slices in the type-reduced set." Because this method does not derive from a type-1 defuzzification method, it is not explained further in this book. An open research question is: How or is this method related to centroid type-reduction for a GT2 FS?

where

$$\tilde{A}_\alpha = \int_{x\in X} \tilde{A}(x)_\alpha / x = \int_{x\in X} [a_\alpha(x), b_\alpha(x)]/x \tag{8.47}$$

Regardless of the kind of type-reduction (centroid, height or center-of-sets), the following is always true (Sect. 7.12): *type-reduction for a GT2 FS (or for more than one GT2 FS) can be viewed as a nonlinear function of the primary variable (or variables) of the set (or sets), and so it can be computed as the fuzzy union of that nonlinear function applied to α-planes.*

8.4.1 Centroid Type-Reduction for GT2 Fuzzy Sets

Theorem 8.4 (Liu 2008) *The centroid of a closed* (Definition 6.25) *GT2 FS* $\tilde{A}$, $C_{\tilde{A}}(x)$, *is a type-1 fuzzy set that can be computed using the horizontal-slice representation of* $\tilde{A}$, *as:*

$$C_{\tilde{A}}(x) = \bigcup_{\alpha\in[0,1]} C_{R_{\tilde{A}_\alpha}}(x) = \bigcup_{\alpha\in[0,1]} \alpha/[c_l(R_{\tilde{A}_\alpha}), c_r(R_{\tilde{A}_\alpha})] \equiv \bigcup_{\alpha\in[0,1]} \alpha/[c_l(\alpha), c_r(\alpha)] \tag{8.48}$$

where $C_{R_{\tilde{A}_\alpha}}(x)$ *is the centroid of the horizontal slice at level* α, $R_{\tilde{A}_\alpha}$ *[see (6.51)].*

Proof Obvious from the horizontal-slice representation of GT2 FS $\tilde{A}$ that is given in (8.46) and (8.47).

Although non-closed GT2 FSs are not covered in this book, it is worth noting that the recent paper (Xie and Lee 2016) has extended the Theorem 8.4 horizontal-slice method for computing the centroid of a closed GT2 FS to non-closed GT2 FSs. Quoting from the abstract of their paper:

> For each decomposed α-plane, we convert it into a group of IT2 FSs. The union of the centroids of its member IT2 FSs constitutes the centroid of the α-plane. Then the weighted union of the centroids of the decomposed α-planes becomes the centroid type-reduced set of the original T2 FS. When dealing with T2 FSs with convex secondary MFs, our proposed method is reduced to the Liu's method.

A procedure for computing $C_{\tilde{A}}(x)$ is [adapted from (Mendel et al. 2009, p. 1195)]:

1. Decide on how many α-planes will be used, where $\alpha \in [0, 1]$. Call that number $k_{\max}$; its choice will depend on the accuracy that is required, if accuracy is important, or its choice will be made a design parameter during the design of a general type-2 fuzzy system.
2. For each α, compute $\tilde{A}_\alpha$.

3. Compute $c_l(\alpha)$ and $c_r(\alpha)$ using two EKM algorithms (Table 8.5) or EIASC algorithms (Table 8.6), or even better by using the monotone centroid flow algorithms that are explained at the end of this section.
4. Repeat Steps 2 and 3 for the $k_{\max}$ values of α chosen in Step 1.
5. Bring all of the $k_{\max}$ $C_{R_{\tilde{A}_\alpha}}(x)$ together using (8.48) to obtain $C_{\tilde{A}}(x)$.

For a closed GT2 FS, because of the convexity of its secondary MFs, the shape of $C_{\tilde{A}}(x)$ is predictable, and the following properties (P) hold (no proofs are given for any of these because they really are quite obvious) (Mendel et al. 2009, p. 1195):

P1. Maximum uncertainty about $\tilde{A}$ occurs for its $\alpha = 0$ plane, $\tilde{A}_{\alpha=0}$.
P2. Minimum uncertainty about $\tilde{A}$ occurs for its $\alpha = 1$ plane, $\tilde{A}_{\alpha=1}$.
P3. When $\alpha_i \geq \alpha_j$, then $C_{\tilde{A}_{\alpha_i}}(x) \subseteq C_{\tilde{A}_{\alpha_j}}(x)$.
P4. $C_{\tilde{A}}(x)$ is first nondecreasing and then nonincreasing.
P5. When all secondary MFs are normal at exactly one point, then $\tilde{A}_{\alpha=1}$ is a *function* (i.e., not a plane) so that $C_{\tilde{A}_1}(x)$ is a single point (i.e., not an interval).
P6. When all secondary MFs are normal triangles, then $C_{\tilde{A}}(x)$ is *triangle-looking*. Its base is computed as $C_{\tilde{A}_0}(x)$ and its apex is computed as $C_{\tilde{A}_1}(x)$. The latter will be a single point (see P5), however its sides may not be straight lines, which is why $C_{\tilde{A}}(x)$ is called "triangle-looking."
P7. When some or all the secondary MFs are normal trapezoids, then $\tilde{A}_{\alpha=1}$ is a plane and $C_{\tilde{A}_1}(x)$ is an interval.
P8. When all the secondary MFs are (normal) trapezoids, then $C_{\tilde{A}}(x)$ is *trapezoid-looking*. Its base is computed as $C_{\tilde{A}_0}(x)$ and its top is computed as $C_{\tilde{A}_1}(x)$. The latter will be an interval (see P7), however, its two sides may not be straight lines, which is why $C_{\tilde{A}}(x)$ is called "trapezoid-looking."
P9. If $\tilde{A}$ is totally symmetrical (see Definition 6.15), then $C_{\tilde{A}}(x)$ is symmetrical about $x = m$, and the mean value (i.e., the defuzzified value) of $C_{\tilde{A}}(x)$ equals m (a proof of this is requested in Exercise 8.16).

These properties about the shape of $C_{\tilde{A}}(x)$ will be confirmed by means of the next two examples. In these examples:

- The domain of the primary variable for $\tilde{A}$ is $x \in [0, 10]$, and x is uniformly sampled using 10,000 samples, such that $x_{i+1} - x_i = 10^{-3}$.
- For α-planes, α is also uniformly sampled so that α has 101 values equal to $0, 1/100, 2/100, \ldots, 99/100, 1$.
- For each α-plane, $C_{\tilde{A}_\alpha}(x)$ is computed by using two EIASC or EKM algorithms.

Example 8.10 [Taken from Mendel et al. (2009, pp. 1196–1197)] (*Gaussian LMF and UMF and trapezoidal secondary MFs*) Shown in Fig. 8.12a is FOU($\tilde{G}$) for

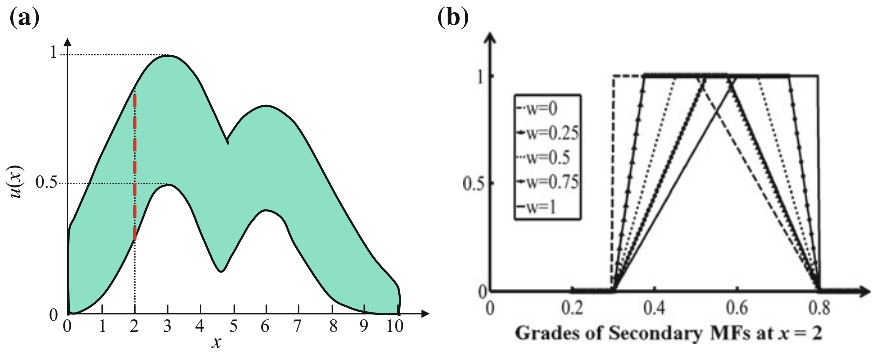


Fig. 8.12 **a** FOU$(\tilde{G})$ for Example 8.10, and **b** secondary MFs at $x = 2$ for w = 0, 0.25, 0.5, 0.75, and 1 (Mendel et al. 2009; © 2009 IEEE)

which $\text{LMF}_{\text{FOU}(\tilde{G})}(x) = \underline{\mu}_{\tilde{G}}(x)$ and $\text{UMF}_{\text{FOU}(\tilde{G})}(x) = \bar{\mu}_{\tilde{G}}(x)$ are each the maximum of two Gaussian functions,[21] i.e.,

$$\underline{\mu}_{\tilde{G}}(x) = \max\left\{0.5\exp\left[-\frac{(x-3)^2}{2}\right], 0.4\exp\left[-\frac{(x-6)^2}{2}\right]\right\} \tag{8.49}$$

$$\bar{\mu}_{\tilde{G}}(x) = \max\left\{\exp\left[-\frac{(x-3)^2}{8}\right], 0.8\exp\left[-\frac{(x-6)^2}{8}\right]\right\} \tag{8.50}$$

In this example, each secondary MF of $\tilde{G}$ is chosen to be a trapezoid whose base equals $\bar{\mu}_{\tilde{G}}(x) - \underline{\mu}_{\tilde{G}}(x)$ and whose top is defined by left and right-end-points, $\text{EP}_l(u|x)$ and $\text{EP}_r(u|x)$, both of which are parameterized as:

$$\text{EP}_l(u|x) = \underline{\mu}_{\tilde{G}}(x) + 0.6w\left[\bar{\mu}_{\tilde{G}}(x) - \underline{\mu}_{\tilde{G}}(x)\right] \tag{8.51}$$

$$\text{EP}_r(u|x) = \bar{\mu}_{\tilde{G}}(x) - 0.6(1-w)\left[\bar{\mu}_{\tilde{G}}(x) - \underline{\mu}_{\tilde{G}}(x)\right] \tag{8.52}$$

where $w = 0, 0.25, 0.5, 0.75, 1$. These secondary MFs are depicted in Fig. 8.12b for $x = 2$. When $w = 0$ the left-hand leg of the trapezoid is vertical,[22] when $w = 0.5$ the trapezoid is symmetrical, and when $w = 1$ the right-hand leg of the trapezoid is vertical.

[21]This FOU is representative of one that might be obtained by computing the union of two fired-rule output sets in a GT2 fuzzy system.

[22]Note that (8.51) and (8.52) are not the same as (6.55) and (6.56), which is why when $w = 0$ (8.51) and (8.52) do not reduce to a square well, whereas (6.55) and (6.56) do.

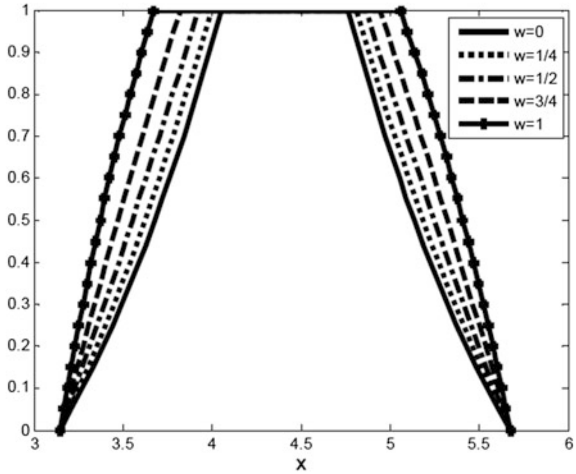


Fig. 8.13 $C_{\tilde{G}}(x)$ for Example 8.10 when w = 0, 0.25, 0.5, 0.75, and 1 (Mendel et al. 2009; © 2009 IEEE)

The centroids of $\tilde{G}$ are depicted in Fig. 8.13 for the five values of w. When $\alpha = 0$, $C_{\tilde{G}_{\alpha=0}}(x) = [3.16, 5.67]$ (this is the support of all five centroids), and therefore the mean value of the two end-points of $C_{\tilde{G}_{\alpha=0}}(x)$, $m(C_{\tilde{G}_{\alpha=0}}(x))$, is approximately (to two significant figures) 4.42.

Table 8.12 provides numerical details for this example for the five values of w. First, it provides the mean value of each centroid, $m(C_{\tilde{G}}(x))$ (i.e., the *defuzzified value* of $\tilde{G}$), obtained by computing the COG of each of the type-1 fuzzy sets in Fig. 8.13, after which it provides the difference and the percentage difference between $m(C_{\tilde{G}_{\alpha=0}}(x))$ and $m(C_{\tilde{G}}(x))$. Surprisingly, the percentage differences are smaller than 1% for all five values of w.

Each of the centroids in Fig. 8.12 looks symmetrical; however, they are not exactly symmetrical, as can be seen from Table 8.12 by comparing $m(C_{\tilde{G}_{\alpha=0}}(x)) = 4.42$ with $m(C_{\tilde{G}}(x))$.

Each centroid was then approximated by a trapezoid that connected only $C_{\tilde{G}_{\alpha=0}}(x)$ and $C_{\tilde{G}_{\alpha=1}}(x)$. Table 8.12 also provides the left-and right-end-points of the

Table 8.12 Example 8.10 results

w	$m(C_{\tilde{G}}(x))$	$m(C_{\tilde{G}_{\alpha=0}}(x)) - m(C_{\tilde{G}}(x))$	Left-end of $C_{\tilde{G}_{\alpha=1}}(x)$	Right-end of $C_{\tilde{G}_{\alpha=1}}(x)$	$m(\hat{C}_{\tilde{G}}(x))$	$m(\hat{C}_{\tilde{G}}(x)) - m(C_{\tilde{G}}(x))$
0	4.41	0.01 (0.23%)	4.06	4.77	4.41	0 (0%)
0.25	4.40	0.02 (0.46%)	4.00	4.81	4.41	0.01 (0.23%)
0.50	4.40	0.02 (0.46%)	3.93	4.87	4.41	0.01 (0.23%)
0.75	4.39	0.03 (0.69%)	3.83	4.95	4.40	0.01 (0.23%)
1	4.38	0.04 (0.92%)	3.68	5.06	4.39	0.01 (0.23%)

Note that $m(C_{\tilde{G}_{\alpha=0}}(x)) \approx 4.42$ [adapted from Table I in Mendel et al. (2009)]

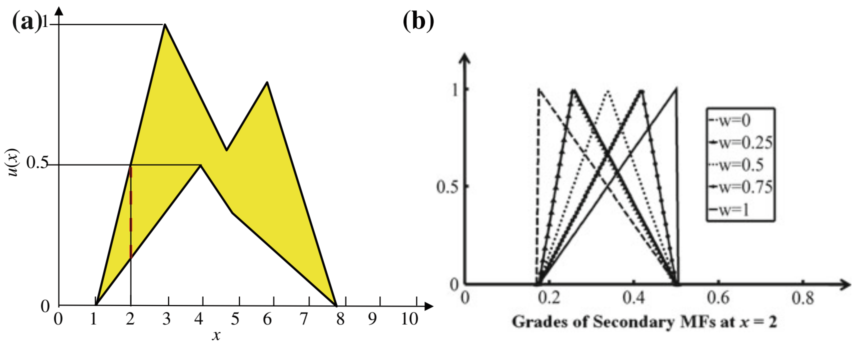


Fig. 8.14 **a** FOU($\tilde{F}$), for Example 8.11, and **b** secondary MFs at $x = 2$ for w = 0, 0.25, 0.5, 0.75, and 1 (Mendel et al. 2009; © 2009 IEEE)

top of each "trapezoidal-looking" centroid, when $\alpha = 1$, the mean value[23] of these approximate centroids, $m(\hat{C}_{\tilde{G}}(x))$, as well as the differences and percentage differences between $m(\hat{C}_{\tilde{G}}(x))$ and $m(C_{\tilde{G}}(x))$. Very surprisingly, the percentage differences between $m(\hat{C}_{\tilde{G}}(x))$ and $m(C_{\tilde{G}}(x))$ for all five values of w are smaller than 1/4%.

These results demonstrate that a very good approximation to the centroid can be obtained by computing centroids for only two α-planes—$\alpha = 0$ and $\alpha = 1$.

Example 8.11 [Taken from Mendel et al. (2009, pp. 1197–1198)] (*Piecewise linear LMF and UMF and triangle*[24] *secondary MFs*) Shown in Fig. 8.14a is FOU($\tilde{F}$) for which $\text{LMF}_{\text{FOU}(\tilde{F})}(x) = \underline{\mu}_{\tilde{F}}(x)$ and $\text{UMF}_{\text{FOU}(\tilde{F})}(x) = \bar{\mu}_{\tilde{F}}(x)$ are each the maximum of two piecewise linear functions, i.e., (Footnote 21 also applies to this FOU)

$$\underline{\mu}_{\tilde{F}}(x) = \left\{ \begin{bmatrix} (x-1)/6 & 1 \le x \le 4 \\ (7-x)/6 & 4 \le x \le 7 \\ 0 & \text{otherwise} \end{bmatrix}, \begin{bmatrix} (x-3)/6 & 3 \le x \le 5 \\ (8-x)/9 & 5 \le x \le 8 \\ 0 & \text{otherwise} \end{bmatrix} \right\} \tag{8.53}$$

$$\bar{\mu}_{\tilde{F}}(x) = \max \left\{ \begin{bmatrix} (x-1)/2 & 1 \le x \le 3 \\ (7-x)/4 & 3 \le x \le 7 \\ 0 & \text{otherwise} \end{bmatrix}, \begin{bmatrix} (x-2)/5 & 2 \le x \le 6 \\ (16-2x)/5 & 6 \le x \le 8 \\ 0 & \text{otherwise} \end{bmatrix} \right\} \tag{8.54}$$

[23]To compute the centroid of a trapezoid, first shift the trapezoid by $-m$ about the origin, i.e., its top must be situated about the origin (it is not necessary for it to be symmetric about the origin), and call the resulting coordinates: $(-b_l, 0)$, $(-a_l, 1)$, $(a_r, 1)$ and $(b_r, 0)$. Compute $C_{\text{Trapezoid|origin}} = \frac{1}{3}[a_r - a_l + b_r - b_l + (a_l b_l - a_r b_r)/(a_r + a_l + b_r + b_l)]$ and then compute $C_{\text{Trapezoid}} = C_{\text{Trapezoid|origin}} + m$.

[24]Starczewski (2009) focuses on triangle secondary MFs. With the advent of Theorem 8.4, his approximations used to compute the centroid for such a GT2 FS are no longer necessary.

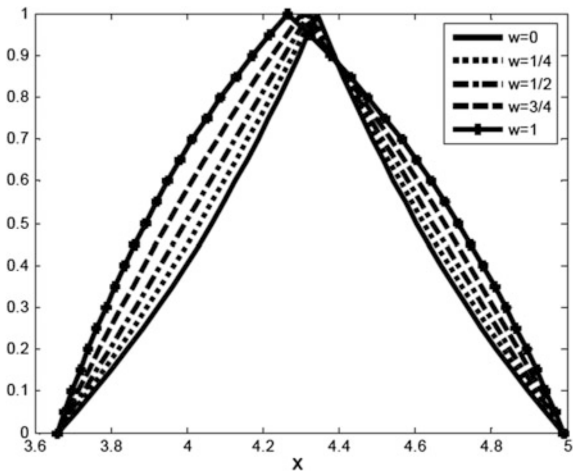


Fig. 8.15 $C_{\tilde{F}}(x)$ for Example 8.11 when w = 0, 0.25, 0.5, 0.75, and 1 (Mendel et al. 2009; © 2009 IEEE)

In this example, each secondary MF of $\tilde{F}$ is chosen to be a triangle of height 1 whose base equals $\bar{\mu}_{\tilde{F}}(x) - \underline{\mu}_{\tilde{F}}(x)$ and whose apex location, $\mathrm{Apex}(u|x)$, is parameterized as:

$$\mathrm{Apex}(u|x) = \underline{\mu}_{\tilde{F}}(x) + w\left[\bar{\mu}_{\tilde{F}}(x) - \underline{\mu}_{\tilde{F}}(x)\right] \tag{8.55}$$

where $w = 0, 0.25, 0.5, 0.75, 1$. These secondary MFs are depicted in Fig. 8.14b for $x = 2$. When $w = 0$ the left-hand leg of the triangle is vertical (so it is a left-sided right triangle), when $w = 0.5$ the triangle is symmetrical, and when $w = 1$ the right-hand leg of the triangle is vertical (so it is a right-sided right triangle).The centroids of $\tilde{F}$ are depicted in Fig. 8.15 for the five values of w. When $\alpha = 0$, $C_{\tilde{F}_{\alpha=0}}(x) = [3.66, 4.99]$ (which is the support of all five centroids), and therefore (to two significant figures) $m(C_{\tilde{F}_{\alpha=0}}(x)) = 4.33$.

Table 8.13 provides numerical details for this example for the five values of w. First, it provides the mean value of centroid, $m(C_{\tilde{F}}(x))$ (i.e., the defuzzified value of $\tilde{F}$) and the difference and percentage difference between $m(C_{\tilde{F}_{\alpha=0}}(x))$ and $m(C_{\tilde{F}}(x))$. Observe that $m(C_{\tilde{F}_{\alpha=0}}(x)) - m(C_{\tilde{F}}(x))$ is smaller than 1% for all five values of w, which is in agreement with Example 8.10.

Each of the centroids in Fig. 8.15 also looks symmetrical; however, they are not exactly symmetrical, as can be seen from Table 8.13 by comparing $m(C_{\tilde{F}_{\alpha=0}}(x)) = 4.33$ with $m(C_{\tilde{F}}(x))$.

Each centroid was then approximated by a triangle that connected $C_{\tilde{F}_{\alpha=0}}(x)$ and $C_{\tilde{F}_{\alpha=1}}(x)$. Table 8.13 also provides the apex location, $C_{\tilde{F}_{\alpha=1}}(x)$, of each of the "triangle-looking" centroids, the mean value[25] of these approximate centroids,

[25]The centroid of triangle $\{(l, 0), (m, h), (r, 0)\}$ is: $C_{\mathrm{Triangle}} = \frac{1}{3}(m + l + r)$. This formula can be used for any triangle, i.e., it does not have to be shifted to the origin.

Table 8.13 Example 8.11 results

w	$m(C_{\tilde{F}}(x))$	$m(C_{\tilde{F}_{\alpha=0}}(x)) - m(C_{\tilde{F}}(x))$	$C_{\tilde{F}_{\alpha=1}}(x)$	$m(\hat{C}_{\tilde{F}}(x))$	$m(\hat{C}_{\tilde{F}}(x)) - m(C_{\tilde{F}}(x))$
0	4.34	−0.01 (−0.23%)	4.34	4.33	−0.01 (−0.23%)
0.25	4.33	0.00 (0%)	4.33	4.33	0 (0%)
0.50	4.32	0.01 (0.23%)	4.32	4.32	0 (0%)
0.75	4.31	0.02 (0.46%)	4.30	4.32	0.01 (0.23%)
1	4.29	0.04 (0.96%)	4.27	4.31	0.02 (0.47%)

Note that $m(C_{\tilde{F}_{\alpha=0}}(x)) \approx 4.33$ [adapted from Table II in Mendel et al. (2009)]

$m(\hat{C}_{\tilde{F}}(x))$, as well as the differences and percentage differences between $m(\hat{C}_{\tilde{F}}(x))$ and $m(C_{\tilde{F}}(x))$. Observe that the percentage differences between $m(\hat{C}_{\tilde{F}}(x))$ and $m(C_{\tilde{F}}(x))$ for all five values of w are smaller than ½%.

This example again demonstrates that a very good approximation to the centroid can be obtained by computing centroids for only two α-planes—$\alpha = 0$ and $\alpha = 1$.

The observations that are made at the end of Examples 8.10 and 8.11 that for triangle and trapezoidal secondary MFs very good approximations to the centroid can be obtained by only using the $\alpha = 0$ and $\alpha = 1$ α-planes, suggests that one does not have to use too many horizontal slices in a GT2 fuzzy system.

If parallel processing is available, the five-step procedure for computing $C_{\tilde{A}}(x)$, that is stated after the proof of Theorem 8.4, can be performed using $2k_{\max}$ processors; however, this five-step procedure does not make use of the additional information that is available about the nature of the secondary MFs of a GT2 FS. The following computationally more efficient procedures for computing $C_{\tilde{A}}(x)$ all make use of this information.

Zhai and Mendel (2011) developed *centroid flow* (*CF*) algorithms in which the first computation is the centroid of the $\alpha = 0$ plane (obtained by using EIASC or EKM algorithms), after which that centroid is propagated up in an iterative manner to the next horizontal slice at level $\Delta\alpha$ using formulas called *centroid flow equations*. The motivation for doing this is the belief that, for small $\alpha_{j+1} - \alpha_j = \Delta\alpha$, the change in the centroid from $R_{\tilde{A}_{\alpha_j}}$ to $R_{\tilde{A}_{\alpha_{j+1}}}$ should also be small, a belief that has been substantiated through numerous simulations.

Unfortunately, as calculations "flow up" from $\alpha = 0$ to $\alpha = 1$ iteration errors accumulate. In order to reduce such errors Zhai and Mendel (2012) provided two improved algorithms, called *enhanced CF* (ECF) algorithms that start at the horizontal slice at level $\alpha = 1/2$. One algorithm moves upwards to the horizontal slice at level $\alpha = 1$, whereas the other algorithm moves downward to the horizontal slice at level $\alpha = 0$. These algorithms reduce the error accumulation by 50%, which may be quite adequate in practical applications of GT2 FSs.

Yeh et al. (2011) have a different kind of flow algorithm. They compute the centroid of the horizontal slice starting at level $\alpha = 1$ using EKM algorithms that are initialized as indicated in Table 8.5, and then that centroid is used to *better* *initialize* the EKM algorithms for the computation of the centroid of the horizontal slice at level $\alpha = 1 - \delta$, after which those results are used to *better initialize* the EKM algorithms for the computation of the centroid of the horizontal slice at level $\alpha = 1 - 2\delta$, etc., until the last centroid is computed for the horizontal slice at level $\alpha = 0$. A reason for beginning with the horizontal slice at level $\alpha = 1$ instead of with the horizontal slice at level $\alpha = 0$ is that, when secondary MFs are all triangles then the FOU of the horizontal slice at level $\alpha = 1$ is a T1 FS, and so one does not need to use EKM algorithms to compute the COG of that set.

Simulations have shown that all of these flow algorithms achieve a 70% reduction in computation time over using EKM algorithms for each horizontal slice at level α-level.

Linda and Manic (2012) developed *monotone centroid flow* (*MCF*) *algorithms* that may not require any EIASC or EKM algorithms, and are to-date the fastest way to compute the centroid of a GT2 FS. The MCF algorithms start with the horizontal slice at level $\alpha = 1$ so as to use an exact COG calculation when all secondary MFs are either triangles (no EIASC or EKM algorithms are needed to do this) or trapezoids[26] (EIASC or EKM algorithms are needed to do this, but only one time). They then move down to the horizontal slice at level $\alpha = 1 - \delta$, but do not use the EKM algorithms at that plane. Instead, they return to *fundamentals*, by focusing on the structure of the solutions in (8.24) and (8.25) but for the horizontal slice at level $\alpha = 1 - \delta$, i.e.,

$$c_l(\tilde{A}_{1-\delta}) = \frac{\sum_{i=1}^{L_{1-\delta}} x_i \bar{\mu}_{\tilde{A}_{1-\delta}}(x_i) + \sum_{i=L_{1-\delta}+1}^{n} x_i \underline{\mu}_{\tilde{A}_{1-\delta}}(x_i)}{\sum_{i=1}^{L_{1-\delta}} \bar{\mu}_{\tilde{A}_{1-\delta}}(x_i) + \sum_{i=L_{1-\delta}+1}^{n} \underline{\mu}_{\tilde{A}_{1-\delta}}(x_i)} \tag{8.56}$$

$$c_r(\tilde{A}_{1-\delta}) = \frac{\sum_{i=1}^{R_{1-\delta}} x_i \underline{\mu}_{\tilde{A}_{1-\delta}}(x_i) + \sum_{i=R_{1-\delta}+1}^{n} x_i \bar{\mu}_{\tilde{A}_{1-\delta}}(x_i)}{\sum_{i=1}^{R_{1-\delta}} \underline{\mu}_{\tilde{A}_{1-\delta}}(x_i) + \sum_{i=R_{1-\delta}+1}^{n} \bar{\mu}_{\tilde{A}_{1-\delta}}(x_i)} \tag{8.57}$$

For convex secondary MFs, it has already been stated above that, when $\alpha_i \geq \alpha_j$ $C_{\tilde{A}_{\alpha_i}}(x) \subseteq C_{\tilde{A}_{\alpha_j}}(x)$; hence, if the centroid at $\alpha = 1$ is $[c_l(\tilde{A}_1), c_r(\tilde{A}_1)]$, then the centroid at $\alpha = 1 - \delta$, $[c_l(\tilde{A}_{1-\delta}), c_r(\tilde{A}_{1-\delta})]$, satisfies the following containment property:

$$[c_l(\tilde{A}_1), c_r(\tilde{A}_1)] \subset [c_l(\tilde{A}_{1-\delta}), c_r(\tilde{A}_{1-\delta})] \tag{8.58}$$

which means:

$$c_l(\tilde{A}_{1-\delta}) < c_l(\tilde{A}_1) \quad \text{and} \quad c_r(\tilde{A}_{1-\delta}) > c_r(\tilde{A}_1) \tag{8.59}$$

[26] In their paper, for the horizontal slice at level $\alpha = 1$, when all secondary MFs are trapezoids, they approximate the COG as the COG of the average value of the lower and upper MFs [as in Nie and Tan (2008)].

By using (8.56)–(8.59), it is easy to find the switch points $L_{1-\delta}$ and $R_{1-\delta}$ from switch points L_1 and R_1 as follows:

1. Compute $c_l(\tilde{A}_1)$ and $c_r(\tilde{A}_1)$ and their associated switch points L_1 and R_1.
2. Express $c_l(\tilde{A}_1)$ as in (8.56) and change the integer L_1 to $L_1 - k$ $(k = 1, \ldots)$.
3. Stop at the first value of k for which $c_l(\tilde{A}_{1-\delta}) > c_l(\tilde{A}_1)$, in which case $L_{1-\delta} = (L_1 - k) + 1$.
4. Compute and save $c_l(\tilde{A}_{1-\delta})$.
5. Similarly, express $c_r(\tilde{A}_1)$ as in (8.57) and change the integer R_1 to $R_1 + k$ $(k = 1, \ldots)$.
6. Stop at the first value of k for which $c_r(\tilde{A}_{1-\delta}) < c_r(\tilde{A}_1)$, in which case $R_{1-\delta} = (R_1 + k) - 1$.
7. Compute and save $c_r(\tilde{A}_{1-\delta})$.

Once $c_l(\tilde{A}_{1-\delta})$ and $c_r(\tilde{A}_{1-\delta})$ have been computed for the horizontal slice at level $\alpha = 1 - \delta$, these steps can be repeated for the horizontal slice at levels $\alpha = 1 - 2\delta, 1 - 3\delta, \ldots, 1 - k_{\max}\delta$, and 0 [where $k_{\max} = (1 - \delta)/\delta$].

Note that, for each horizontal slice at level α, Linda and Manic are in effect using the EKMANI idea (Wu and Nie 2011) of using the just-computed centroid to initialize the computation of the next centroid.

Because of the relative simplicity of the MCF algorithms over the other flow algorithms, as well as their computational advantages, it is the MCF algorithms that are recommended by this author.

Example 8.12 Linda and Manic (2012) have many very interesting computational results in their paper. Only some of them are mentioned here. All are for the GT2 FSs $\tilde{G}$ and $\tilde{F}$ that have been described in Examples 8.10 and 8.11, respectively. To begin, they compared the *accuracy* obtained by using the KM and MCF algorithms[27] when $w = \{0.1, 0.9\}$, and concluded: "… the MCF and KM algorithms compute numerically identical solutions."

They then studied *computation times* for KM, EKM, CF, and MCF algorithms, for many values of n (samples, ranging from 20 to 215) and $k_{\max}$ (number of horizontal slices, ranging from 2 to 100), and for $w = \{0.25, 0.5, 0.75, 1\}$. The algorithms were executed on Dell Precision M4500, Intel Core i7 CPU Q720 @ 1.60 GHz with 8 GB RAM, running Windows 7 in a Matlab environment. Computation times were averaged over all n, $k_{\max}$ and w. The overall average speed-up of MCF over KM was approximately 85% for both $\tilde{G}$ and $\tilde{F}$, whereas it was approximately 55–58% for EKM over KM.[28]

---

[27]They also do this for the CF algorithms, but these results are not included here because CF algorithms are too complicated. Also, they only consider the KM algorithms for their accuracy study because exactly the same numerical results are obtained from the EKM algorithms.

[28]For CF over KM, it was 88–90% for $\tilde{F}$, but only 67–69% for $\tilde{G}$.

8.4.2 Centroid Type-Reduction in a GT2 Fuzzy System

When an input $\mathbf{x} = \mathbf{x}'$ is applied to a general type-2 rule (i.e., a rule that looks just like a type-1 rule except that some or all of its fuzzy sets are GT2 FSs) it leads to a type-1 fuzzy *firing set* $F(\mathbf{x}')$ which may then be combined with the entire GT2 consequent of that rule, $\tilde{G}^l$, by means of the meet operation, leading to a *fired-rule output GT2 FS*, $\tilde{B}^l$. Then, all of the fired-rule output GT2 FSs may be combined by means of the join operation, producing one *combined fired-rule GT2 output fuzzy set*, $\tilde{B}$. $\tilde{B}$ is then *type-reduced* by computing its centroid to give $Y_c(\mathbf{x}')$, which is computed as explained in Sect. 8.4.1. So, Theorem 8.4 applies directly to centroid type-reduction in a GT2 fuzzy system. This is explained in Chap. 11, but using horizontal slices from the very beginning.

8.4.3 COS Type-Reduction in a GT2 Fuzzy System

Center-of-sets (COS) type-reduction is explained in Sect. 11.6.2, and uses the horizontal-slice representation for a GT2 FS. Height type-reduction is left as an exercise for the reader (Exercise 11.4) because as of the writing of this book (2017) COS type-reduction is being used in applications, whereas height type-reduction is not.

Appendix 1: A Wavy-Slice Approach to Type-Reduction

The original approaches to all kinds of type-reduction were based on the Wavy-Slice Representation Theorem 6.2, even though, as has been mentioned in the main body of this chapter, this theorem did not exist when the type-reduction methods were invented. This appendix summarizes the wavy-slice approach to centroid and COS type-reduction. A reader who is interested in height type-reduction can find similar results in Mendel (2001, Sect. 9.5.3).

1.1 Centroid Type-Reduction

The starting point for centroid type-reduction is a GT2 FS (or an IT2 FS) $\tilde{B}$ and its MF $\mu_{\tilde{B}}(y, u)$, where $y \in Y$. The centroid, $C_{\tilde{B}}(y)$, is defined (or obtained by using the Extension Principle[29]) as the union of the centroids of all the embedded T2 FSs of $\tilde{B}$, i.e.,

---

[29] $\tilde{B}$ is also the union of all of its vertical slices (secondary MFs), each of which is a T1 FS. In $\sum_{i=1}^{N} y_i\theta_i / \sum_{i=1}^{N} \theta_i$, which appears in (8.60), each θ_i $(i = 1, \ldots, N)$ is chosen from the respective support of one of these T1 FSs [each with membership grade $f_{y_i}(\theta_i)$], and so $\sum_{i=1}^{N} y_i\theta_i / \sum_{i=1}^{N} \theta_i$ is a function of N T1 FSs; hence, its MF is found from the Extension Principle.

$$C_{\tilde{B}}(y) = \int\limits_{\theta_1 \in I_{y_1}} \cdots \int\limits_{\theta_N \in I_{y_N}} \left[\min_{i=1,\ldots,N} \{ f_{y_i}(\theta_i) \} \right] \Bigg/ \frac{\sum_{i=1}^{N} y_i \theta_i}{\sum_{i=1}^{N} \theta_i} \tag{8.60}$$

The initial procedure for computing $C_{\tilde{B}}(y)$, stated next, assumes that $\mu_{\tilde{B}}(y,u)$ is known (how to compute it for an IT2 fuzzy system is explained in Chap. 9, and how to compute it for a GT2 fuzzy system is explained in Chap. 11):

1. Discretize Y into N points $y_1, \ldots, y_N$.
2. Discretize each I_{y_i} (the support of the secondary MF at y_i) into a suitable number of points, say M_i $(i = 1, \ldots, N)$. Let $\theta_i \in I_{y_i}$.
3. Enumerate all the embedded T1 FSs of $\tilde{B}$; there will be $\prod_{i=1}^{N} M_i$ of them.
4. Compute the centroid of each enumerated embedded T1 FS and assign it a membership grade equal to the minimum of the secondary grades corresponding to that enumerated embedded T1 FS.

Mathematically, this means Mendel (2007, Sect. 2.4):

$$C_{\tilde{B}}(y) = \left\{ \left[\zeta_k, \left(\min_{i=1,\ldots,N} \{ f_{y_i}(\theta_i) \} \right)_k \right] \right\}_{k=1}^{\prod_{i=1}^{N} M_i} \tag{8.61}$$

$$\zeta_k = \left(\frac{\sum_{i=1}^{N} y_i \theta_i}{\sum_{i=1}^{N} \theta_i} \right)_{k^{\text{th}}\text{embedded T1 FS}} \tag{8.62}$$

Although not stated explicitly in (8.61), if two or more embedded T1 FSs have the same centroid, the one with the largest value of $\min_{i=1,\ldots,N}\{f_{y_i}(\theta_i)\}$ is kept. Note, also, that in (8.61) the minimum t-norm must be used (and not the product t-norm), because the minimum t-norm is unaffected by a change in N, whereas the product t-norm changes as N changes (Karnik and Mendel 2001, Sect. 2.1; Mendel 2001, p. 252).

This four-step procedure is not very practical because it requires the explicit enumeration of the $\prod_{i=1}^{N} M_i$ embedded T1 FSs of $\tilde{B}$, so that their centroids can be computed, and this number will in general be extremely large. For IT2 FSs it was replaced in Karnik and Mendel (2001) and Mendel (2001) by KM algorithms. However, it was not replaced for GT2 FSs until Liu (2008) showed how $C_{\tilde{B}}(y)$ could be computed by first expressing $\tilde{B}$ using its horizontal-slice representation.

A novel way for computing $C_{\tilde{B}}(y)$ (Greenfield et al. 2005) is motivated by the fact that two embedded T2 FSs can be very alike, and is based on randomly sampling the embedded T2 FSs and computing their centroids. Doing this gives rise to a significant reduction in the time or resources needed to perform centroid type-reduction. Greenfield et al. (2005) has examples (for four different primary MFs and different discretizations of them) that demonstrate that the number of randomly selected embedded T2 FSs only marginally affects the COG of $C_{\tilde{B}}(y)$ (the defuzzified output). Excellent results have been obtained for as few as 10 randomly selected embedded T2 FSs.

1.2 COS Type-Reduction

The starting point for COS type-reduction is the so-called Generalized Centroid, *GC*, where:

$$\mathrm{GC} = \int\limits_{z_1 \in Z_1} \cdots \int\limits_{z_n \in Z_n} \int\limits_{w_1 \in W_1} \cdots \int\limits_{w_n \in W_n} \left[T_{i=1}^{n} \mu_{Z_i}(z_i) \bigstar T_{i=1}^{n} \mu_{W_i}(w_i) \right] \Bigg/ \frac{\sum_{i=1}^{n} z_i w_i}{\sum_{i=1}^{n} w_i} \tag{8.63}$$

The initial procedure for computing GC, stated next, assumes that the $2n$ MFs $\mu_{Z_i}(z_i)$ and $\mu_{W_i}(w_i)$ $(i = 1, \ldots, n)$ are known (how to compute them for an IT2 fuzzy system is explained in Chap. 9, and how to compute them for a GT2 fuzzy system is explained in Chap. 11):

1. Discretize the domain of each T1 FS Z_i into a suitable number of points, say N_i $(i = 1, \ldots, n)$.
2. Discretize the domain of each T1 FS W_i into a suitable number of points, say M_i $(i = 1, \ldots, n)$.
3. Enumerate all the possible combinations $\boldsymbol{\theta} = [z_1, \ldots, z_n, w_1, \ldots, w_n]^T$ such that $z_i \in Z_i$ and $w_i \in W_i$. The total number of combinations will be $\prod_{i=1}^{n} M_i N_i$.
4. Compute the centroid $\sum_{i=1}^{n} z_i w_i \big/ \sum_{i=1}^{n} w_i$ of each of the enumerated combinations and assign it a membership grade equal to the t-norm $T_{i=1}^{n} \mu_{Z_i}(z_i) \bigstar T_{i=1}^{n} \mu_{W_i}(w_i)$.

Mathematically, this means (Mendel 2007):

$$\mathrm{GC} = \left\{ \left[\xi_k, \left(T_{i=1}^{n} \mu_{Z_i}(z_i) \bigstar T_{i=1}^{n} \mu_{W_i}(w_i) \right)_k \right] \right\}_{k=1}^{\prod_{i=1}^{n} M_i N_i} \tag{8.64}$$

$$\xi_k = \left(\frac{\sum_{i=1}^{n} z_i w_i}{\sum_{i=1}^{n} w_i} \right)_{k^{\text{th}}\text{combination}} \tag{8.65}$$

Although not stated explicitly in (8.64), if two or more ξ_k have the same value, the one with the largest value of $T_{i=1}^{n} \mu_{Z_i}(z_i) \bigstar T_{i=1}^{n} \mu_{W_i}(w_i)$ is kept. Note that, when the *GC* is computed in a type-2 fuzzy system, $n = M$, where M is the number of rules and is *fixed*, hence product or minimum t-norm can be used in (8.64).

This four-step procedure is not very practical because it requires the explicit enumeration of $\prod_{i=1}^{n} M_i N_i$ combinations of $[z_1, \ldots, z_n, w_1, \ldots, w_n]^T$ so that their weighted average can be computed, and this number is also in general extremely large.

The GC in (8.62) is now recognized to be an IWA and it can be computed as explained in Sect. 8.2.

Appendix 2: Type-Reduction Properties

This appendix presents theoretical properties about the IWA, states the Continuous KM algorithms (for computing the centroid of an IT2 FS) and presents their properties. All of the properties in this appendix will help one to better understand the rapid convergence of KM and EKM algorithms.

2.1 Properties of the IWA

The following three properties are for computing y_l in the IWA and appeared first in Liu and Mendel (2008) (comparable properties for y_r are stated after these three properties). They assume that a_i and b_i have been sorted in increasing order, i.e., $a_1 \leq a_2 \leq \cdots \leq a_n$ and $b_1 \leq b_2 \leq \cdots \leq b_n$.

Property 8.6 [Location Property for $y_l(L)$] *When $k = L$, then*

$$a_L \leq y_l(L) = y_l < a_{L+1} \tag{8.66}$$

where $y_l(L)$ is defined in (8.15). In (8.66), it is assumed that $a_{L+1} \neq a_L$. If, however, $a_{L+1} = a_L$, then change the right-hand side of (8.66) to: "the first value of i such that $a_i \neq a_L$."

This property locates $y_l(L)$ either between two specific adjacent values of a_i, or at the left-end-point of these adjacent values [see where $y_l(L)$ is located on the vertical axis of Fig. 8.16, and note that if $a_{L+1} = a_L$, then a_L and a_{L+1} will lie on top of one another]. When $a_i = b_i \equiv x_i$ (e.g., when the IWA is used to compute the centroid of an IT2 FS), and $x_{i+1} - x_i \rightarrow 0$, then (8.66) implies that $y_l = x_L$. That this is correct is explained in Sect. 2.2.

Proof of the Property 8.6 [Adapted from Liu and Mendel (2008, p. 10)] That (8.66) is true follows directly from Step 2 of the KM and EKM algorithms, when $k = L$, i.e., Step 2 is always enforced when the algorithms stop and convergence has occurred.

Property 8.7 [Shape Property for $y_l(k)$; location of $y_l(k)$ in relation to the line $y = a_k$] *$y_l(k)$ lies above the line $y = a_k$ when a_k is below y_l and lies below the line $y = a_k$ when a_k is above y_l, i.e.,*

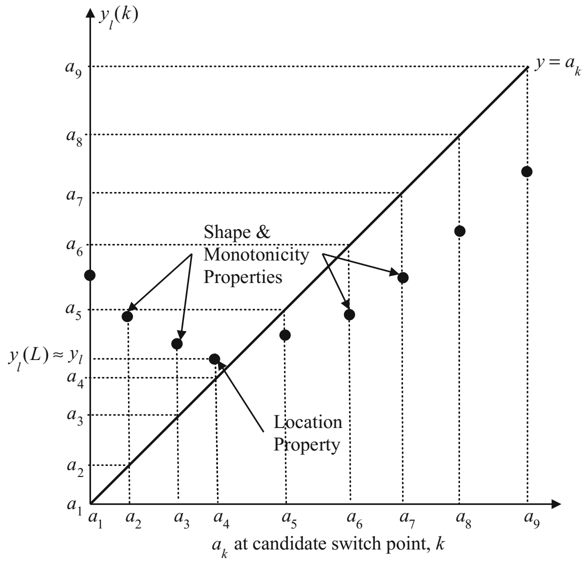


Fig. 8.16 Illustration of the three properties associated with finding $y_l(L)$. The *solid line* shown for $y = a_k$ only has values at $a_1, \ldots, a_9$; and the *large dots* are $y_l(k)$ in (8.17) for $k = 1, \ldots, 9$. The 45° line $y = a_k$ is shown because of the computations in Step 2 of the KM (or EKM) algorithm for y_l (Mendel et al. 2014, p. 59; © 2014 IEEE)

$$\begin{cases} y_l(k) > a_k & \text{when } \; a_k < y_l \\ y_l(k) < a_k & \text{when } \; a_k > y_l \end{cases} \tag{8.67}$$

The Shape Property explains the shape of $y_l(k)$ both to the left and right of its minimum point (see the large dots on Fig. 8.16).

Proof of Property 8.7 [Adapted from Liu and Mendel (2008, p. 10)] Because y_l is the minimum of $y_l(k)$, it must be true that

$$y_l(k) \geq y_l, \quad \text{for } \; k = 1, \ldots, n \tag{8.68}$$

When $y_l > a_k$, (8.68) can be expressed as:

$$y_l(k) \geq y_l > a_k \tag{8.69}$$

Consequently, $y_l(k) > a_k$ when $a_k < y_l$, which completes the proof of the first row of (8.67).

From (8.15), observe that:

$$y_l \left(\sum_{i=1}^{L} d_i + \sum_{i=L+1}^{n} c_i \right) = \sum_{i=1}^{L} a_i d_i + \sum_{i=L+1}^{n} a_i c_i \tag{8.70}$$

Let $k \geq L+1$ and add $\sum_{i=L+1}^{k} a_i(d_i - c_i)$ to both sides of (8.70), to see that:

$$
\begin{aligned}
y_l\left(\sum_{i=1}^{L} d_i + \sum_{i=L+1}^{n} c_i\right) &+ \sum_{i=L+1}^{k} a_i(d_i - c_i) \\
&= \sum_{i=1}^{L} a_i d_i + \sum_{i=L+1}^{n} a_i c_i + \sum_{i=L+1}^{k} a_i(d_i - c_i) \\
&= \sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i
\end{aligned} \tag{8.71}
$$

Because $k \geq L+1$ and (8.66) is true, it must also be true that $y_l < a_k$. Applying this fact and the fact that $d_i - c_i \geq 0$ (for all i) to the left-hand side of (8.71), it follows that (8.71) leads to the following inequality:

$$
\begin{aligned}
\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i &< a_k\left(\sum_{i=1}^{L} d_i + \sum_{i=L+1}^{n} c_i\right) + a_k \sum_{i=L+1}^{k} (d_i - c_i) \\
&< a_k\left(\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i\right)
\end{aligned} \tag{8.72}
$$

Because $\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i > 0$, (8.72) can also be expressed as

$$
a_k > \frac{\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i}{\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i} = y_l(k) \tag{8.73}
$$

Combining the already stated $y_l < a_k$ and (8.73), one concludes that

$$
y_l(k) < a_k \quad \text{when} \quad a_k > y_l \tag{8.74}
$$

This completes the proof of the second row of (8.67).

Property 8.8 [Monotonicity Property of $y_l(k)$] *It is true that:*

$$
\begin{cases} y_l(k-1) \geq y_l(k) & \text{when} \quad a_k < y_l \\ y_l(k+1) \geq y_l(k) & \text{when} \quad a_k > y_l \end{cases} \tag{8.75}
$$

The *Monotonicity* Property also helps us to understand the shape of $y_l(k)$. When $y_l(k)$ is going in the downward direction (see Fig. 8.16) it cannot change that direction before $a_k = y_l$; and, after $a_k = y_l$, when it goes in the upward direction it cannot change that direction.

Proof of Property 8.8 [Adapted from Liu and Mendel (2008, p. 11)] To begin, observe that when $a_k < y_l$ (8.67) lets one conclude that $y_l(k) > a_k$; hence, using (8.15) in this inequality, it is true that

$$\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i > a_k \left(\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i \right) \tag{8.76}$$

This inequality is used below.

First, a formula for $y_l(k) - y_l(k-1)$ is obtained:

$$\begin{aligned}
y_l(k) - y_l(k-1) &= \frac{\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i}{\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i} - \frac{\sum_{i=1}^{k-1} a_i d_i + \sum_{i=k}^{n} a_i c_i}{\sum_{i=1}^{k-1} d_i + \sum_{i=k}^{n} c_i} \\
y_l(k) - y_l(k-1) &= \frac{\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i}{\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i} - \frac{\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i - a_k(d_k - c_k)}{\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i - (d_k - c_k)} \\
y_l(k) - y_l(k-1) &= \frac{(d_k - c_k)\left[a_k\left(\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i\right) - \left(\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i\right)\right]}{\left(\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i\right)\left(\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i - (d_k - c_k)\right)}
\end{aligned} \tag{8.77}$$

Note that going from the second to the third lines of (8.77) has involved a moderate amount of straightforward algebra during which many terms cancel each other. The denominator of (8.77) is positive. In its numerator: $d_k - c_k \geq 0$, $\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i > 0$, and from (8.76) the second term in the bracket is always larger than the first term in the bracket; hence, the numerator of (8.76) ≤ 0, which therefore means, that:

$$y_l(k) - y_l(k-1) \leq 0 \tag{8.78}$$

Combining the fact that $a_k < y_l$ with (8.78) completes the proof of the first line of (8.75).

Next, consider the case when $a_k > y_l$, for which (8.67) lets one conclude

$$\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i < a_k \left(\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i \right) \tag{8.79}$$

Because the rest of the proof of the second line of (8.75) is so similar to that just given for the first line, its details are left to the reader (Exercise 8.17).

The comparable three properties for y_r are stated next without proofs (the proofs, as well as a figure that is analogous to Fig. 8.16, are asked for in Exercise 8.18). They also appeared first in Liu and Mendel (2008).

Property 8.9 [Location Property for $y_r(R)$] *When* $k = R$, *then*

$$b_R \leq y_r(R) = y_r < b_{R+1} \tag{8.80}$$

where $y_r(R)$ *is defined in* (8.16). *In* (8.80)*, it is assumed that* $b_{R+1} \neq b_R$*. If, however,* $b_{R+1} = b_R$*, then change the right-hand side of* (8.80) *to: "the first value of* i *such that* $b_i \neq b_R$*."*

Property 8.10 [Shape Property for $y_r(k)$; location of $y_r(k)$ in relation to the line $y = b_k$] $y_r(k)$ *lies above the line* $y = b_k$ *when* b_k *is below* y_r *and lies below the line* $y = b_k$ *when* b_k *is above* y_r*, i.e.,*

$$\begin{cases} y_r(k) > b_k & \text{when } b_k < y_r \\ y_r(k) < b_k & \text{when } b_k > y_r \end{cases} \tag{8.81}$$

Property 8.11 [Monotonicity Property of $y_r(k)$] *It is true that:*

$$\begin{cases} y_r(k-1) \leq y_r(k) & \text{when } b_k < y_r \\ y_r(k+1) \leq y_r(k) & \text{when } b_k > y_r \end{cases} \tag{8.82}$$

From knowledge of the shapes[30] of $y_l(k)$ and $y_r(k)$, it should be clear to readers who are familiar with optimization theory that the two optimization problems in (8.5) and (8.6) are *easy*. Each problem has only one global extremum (Fig. 8.16) and there are no local extrema. Regardless of how one initializes any algorithm for finding the extremum, convergence will occur, i.e., it is impossible to become trapped at a local extremum because of how each algorithm is initialized. It is also obvious, from the shape of $y_l(k)$ [and $y_r(k)$] that the algorithm that computes $y_l(L)$ [and $y_r(R)$] will converge very quickly. In fact, the shape of $y_l(k)$ [and $y_r(k)$] suggests that quadratic convergence should be possible.

2.2 Continuous KM Algorithms for the Centroid of an IT2 FS and Its Properties

Some interesting properties have been developed about the centroid of an IT2 FS $\tilde{A}$ and the convergence rates of the KM algorithms for the situation when one begins with mathematical formulas for the lower and upper MFs, so that the universe of discourse for $\tilde{A}$ can be treated as continuous. An alternative interpretation is to let the discretization size (sampling size) approach 0. This work has led to the continuous KM (CKM) algorithms that are summarized in Table 8.14.

The following properties are only stated; their proofs can be found in the appropriate references.

[30] Such shape knowledge only became known in 2008 (Liu and Mendel 2008); it was not known when the KM algorithms were invented.

Table 8.14 Continuous KM (CKM) algorithms[a] to compute the centroid end-points of an IT2 FS, $\tilde{A}$ (Liu and Mendel 2011)

Step	CKM algorithm for c_l	CKM algorithm for c_r
	$c_l = \min_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\bar{\mu}_{\tilde{A}}(x)]} \frac{\int_a^b x\theta(x)dx}{\int_a^b \theta(x)dx}$	$c_r = \max_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\bar{\mu}_{\tilde{A}}(x)]} \frac{\int_a^b x\theta(x)dx}{\int_a^b \theta(x)dx}$
1	Let $\theta(x) = [\underline{\mu}_{\tilde{A}}(x) + \bar{\mu}_{\tilde{A}}(x)]/2$ and compute the initial value ξ, as $\xi = \int_a^b x\theta(x)dx \Big/ \int_a^b \theta(x)dx$	
2	Compute $\xi_l = \frac{\int_a^\xi x\bar{\mu}_{\tilde{A}}(x)dx + \int_\xi^b x\underline{\mu}_{\tilde{A}}(x)dx}{\int_a^\xi \bar{\mu}_{\tilde{A}}(x)dx + \int_\xi^b \underline{\mu}_{\tilde{A}}(x)dx}$	Compute $\xi_r = \frac{\int_a^\xi x\underline{\mu}_{\tilde{A}}(x)dx + \int_\xi^b x\bar{\mu}_{\tilde{A}}(x)dx}{\int_a^\xi \underline{\mu}_{\tilde{A}}(x)dx + \int_\xi^b \bar{\mu}_{\tilde{A}}(x)dx}$
3	Check if $\lvert\xi - \xi_l\rvert \le \varepsilon$ (ε is a given error bound of the algorithm). If yes, stop and set $c_l = \xi_l$. If no, go to Step 4	Check if $\lvert\xi - \xi_r\rvert \le \varepsilon$ (ε is a given error bound of the algorithm). If yes, stop and set $c_r = \xi_r$. If no, go to Step 4
4	Set $\xi = \xi_l$ and go to Step 2	Set $\xi = \xi_r$ and go to Step 2

[a]Continuous EKM algorithms are given in Mendel (2013, Table V)

Property 8.12 (Liu and Mendel 2011) *The continuous versions of* (8.17) and (8.19), *for finding the centroid of an IT2 FS, are:*

$$c_l = \min_{\xi\in[a,b]} c_l(\xi) = \min_{\xi\in[a,b]} \frac{\int_a^\xi x\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x + \int_\xi^b x\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x}{\int_a^\xi \bar{\mu}_{\tilde{A}}(x)\mathrm{d}x + \int_\xi^b \underline{\mu}_{\tilde{A}}(x)\mathrm{d}x} \tag{8.83}$$

$$c_r = \max_{\xi\in[a,b]} c_r(\xi) = \max_{\xi\in[a,b]} \frac{\int_a^\xi x\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x + \int_\xi^b x\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x}{\int_a^\xi \underline{\mu}_{\tilde{A}}(x)\mathrm{d}x + \int_\xi^b \bar{\mu}_{\tilde{A}}(x)\mathrm{d}x} \tag{8.84}$$

Property 8.13 (Mendel and Wu 2006, 2007) *c_l and c_r are the solutions to the following integral equations:*

$$c_l = \frac{\int_a^{c_l} x\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x + \int_{c_l}^b x\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x}{\int_a^{c_l} \bar{\mu}_{\tilde{A}}(x)\mathrm{d}x + \int_{c_l}^b \underline{\mu}_{\tilde{A}}(x)\mathrm{d}x} \tag{8.85}$$

$$c_r = \frac{\int_a^{c_r} x\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x + \int_{c_r}^b x\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x}{\int_a^{c_r} \underline{\mu}_{\tilde{A}}(x)\mathrm{d}x + \int_{c_r}^b \bar{\mu}_{\tilde{A}}(x)\mathrm{d}x} \tag{8.86}$$

This property is another Location Property and is in agreement with the Location Properties 8.6 and 8.9 in Sect. 2.1 of this Appendix. Loosely speaking, it states that c_l and c_r each correspond to their switch point.

One approach to finding c_l and c_r is to solve for them from (8.85) and (8.86). If, for example, $\underline{\mu}_{\tilde{A}}(x)$ and $\bar{\mu}_{\tilde{A}}(x)$ are linear functions of x, then an examination of these equations reveals that the resulting equations for c_l and c_r are *cubic* in c_l or c_r.

Usually, there are no closed-form formulas that can be found for c_l and c_r from these equations. One exception to this is given in the following:

Example 8.13 (Mendel and Wu 2007) A fuzzy (information) granule (Definition 6.16) is depicted in Fig. 8.17. It is possible to obtain the following closed-form formulas for it by using (8.85) and (8.86) (Exercise 8.19):

$$c_l(\tilde{A}) = \frac{\sqrt{L}a_R + \sqrt{R}a_L}{\sqrt{R} + \sqrt{L}} \tag{8.87}$$

$$c_r(\tilde{A}) = \frac{\sqrt{R}a_R + \sqrt{L}a_L}{\sqrt{R} + \sqrt{L}} \tag{8.88}$$

These closed-form formulas were possible because $\underline{\mu}_{\tilde{A}}(x)$ and $\bar{\mu}_{\tilde{A}}(x)$ are constants, so that the equations for c_l and c_r are quadratic.

Property 8.14 (Liu and Mendel 2011) *The optimal solutions of* (8.83) *and* (8.84) *can be transformed into root-finding problems, i.e.,: (a)* $c_l = c_l(\xi^*)$ *is the unique minimum value of* (8.83) *and* ξ^* *is the unique simple root of the monotonic increasing convex function* $\varphi(\xi)$, *where*

$$\varphi(\xi) = \int_a^{\xi} (\xi - x)\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x + \int_{\xi}^{b} (\xi - x)\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x \tag{8.89}$$

(b) $c_r = c_r(\xi^*)$ *is the unique maximum value of* (8.84) *and* ξ^* *is the unique simple root of the monotonic decreasing convex function* $\psi(\xi)$, *where*

$$\psi(\xi) = -\int_a^{\xi} (\xi - x)\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x - \int_{\xi}^{b} (\xi - x)\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x \tag{8.90}$$

As a result of this property, one can apply methods for solving nonlinear root-finding problems to (8.89) and (8.90), e.g., doing this for the Newton–Raphson algorithm, one finds:

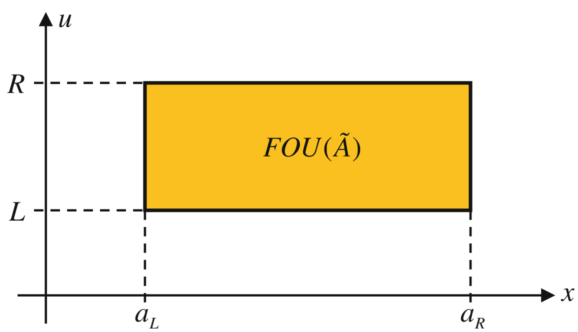


Fig. 8.17 FOU of an IT2 fuzzy granule (Mendel and Wu 2007, © Elsevier 2007)

Property 8.15 (Liu and Mendel 2011)[31] *The iteration processes in the CKM algorithms in* Table 8.14 *are equivalent to the Newton–Raphson root-finding method* (Tjalling 1995) *for solving* $\varphi(\xi) = 0$ *and* $\psi(\xi) = 0$ *with the iteration formulae:*

$$\xi_{k+1} = \xi_k - \frac{\varphi(\xi_k)}{[\partial\varphi(\xi)/\partial\xi]_{\xi=\xi_k}} \tag{8.91}$$

$$\xi_{k+1} = \xi_k - \frac{\psi(\xi_k)}{[\partial\psi(\xi)/\partial\xi]_{\xi=\xi_k}} \tag{8.92}$$

Property 8.16 (Liu and Mendel 2011) *The convergence speed of the CKM algorithms is quadratic.*[32]

This follows from the well-known result (Tjalling 1995) that Newton–Raphson algorithms converge quadratically.[33] Property 8.16 helps to explain the rapid convergence of KM and EKM algorithms when they are used to compute IWAs, even though CKM algorithms are only applicable for computing the centroid of an IT2 FS.

Property 8.17 (Han and Liu 2016) *CKM algorithms exhibit global convergence, i.e., for any initialization method the CKM algorithms will always converge to the sample optimal solutions* c_l *and* c_r.

This is also evident for the KM (EKM) algorithms from Fig. 8.16.

Finally, it is important that the reader be reminded that the results in this appendix only apply to the situation where an FOU exists, i.e., for centroid type-reduction They do not apply to height or COS type-reduction.

Exercises

8.1 Explain why (8.18) and (8.20) are valid representations of y_l and y_r that are given in (8.15) and (8.16), respectively.

8.2 For the following data, compute y_l and y_r by using KM algorithms. Show the computations in tables like those in Example 8.1.

[31]The wording of this property that is given here is taken from Han and Liu (2016).

[32]An earlier convergence result that is in Mendel and Liu (2007) is: the CKM algorithms are super-exponentially convergent, meaning that convergence is not linear. The significance of Property 8.16 is that it quantifies "super-exponential" as quadratic.

[33]Other optimization methods can also be used to solve $\varphi(\xi) = 0$ and $\psi(\xi) = 0$, e.g., Halley's Method (Alefeld 1981), which is cubically convergent, but is much more computationally intensive. See, also, Liu and Mendel (2011).

i	$[a_i, b_i]$	$[c_i, d_i]$
1	[8.2, 9.8]	[1.0, 3.0]
2	[5.8, 8.2]	[0.6, 1.4]
3	[2.0, 8.0]	[7.1, 8.9]
4	[3.0, 5.0]	[2.4, 5.6]
5	[0.5, 1.5]	[5.0, 7.0]

8.3 Repeat Exercise 8.2 using the EKM algorithms.

8.4 Repeat Exercise 8.2 using the EIASC algorithms. Show the computations in tables like those in Example 8.4.

8.5 Using the data that are given in Exercise 8.2:

(a) Find switch points L and R.

(b) Create figures that are analogous to Figs. 8.2 and 8.3.

(c) How many switches occur from c_i to d_i in the unordered W_i for $y_l(L)$, and where do they occur?

(d) How many switches occur from d_i to c_i in the unordered W_i for $y_r(R)$, and where do they occur?

8.6 Given the FOU of an IT2 FS that is symmetrical about the primary variable x at $x = m$. Prove that once c_l has been found c_r can be computed as $c_r = 2m - c_l$. It is okay to use Item 1 on p. 406 to do this.

8.7 Prove Property 8.2 for a continuous universe of discourse.

8.8 Prove Property 8.3.

8.9 Explain how (or if) Property 8.5 can be applied to the EIASC.

8.10 Prove that renumbering M rules does not change $y_l^h(\mathbf{x}')$ and $y_r^h(\mathbf{x}')$, given in (8.34) and (8.35), respectively.

8.11 Prove that renumbering M rules does not change $y_l^{\text{COS}}(\mathbf{x}')$ and $y_r^{\text{COS}}(\mathbf{x}')$, given in (8.43) and (8.44), respectively.

8.12 (Wu 2012) Compute $y_l^{\text{COS}}(\mathbf{x}')$ and $y_r^{\text{COS}}(\mathbf{x}')$ when: $M = 4$, $[\underline{f}^1(\mathbf{x}'), \bar{f}^1(\mathbf{x}')] = [0.22, 0.90]$, $[\underline{f}^2(\mathbf{x}'), \bar{f}^2(\mathbf{x}')] = [0, 0.405]$, $[\underline{f}^3(\mathbf{x}'), \bar{f}^3(\mathbf{x}')] = [0.055, 0.60]$, $[\underline{f}^4(\mathbf{x}'), \bar{f}^4(\mathbf{x}')] = [0, 0.27]$, and rule consequents are T1 FSs, with $c(G^1) = -2.29$, $c(G^2) = -1.88$, $c(G^3) = 1.88$ and $c(G^4) = 2.29$. What are L and R?

8.13 (Wu 2012) Compute $y_l^{\text{COS}}(\mathbf{x}')$ and $y_r^{\text{COS}}(\mathbf{x}')$ when: $M = 4$, $[\underline{f}^1(\mathbf{x}'), \bar{f}^1(\mathbf{x}')] = [0, 0.405]$, $[\underline{f}^2(\mathbf{x}'), \bar{f}^2(\mathbf{x}')] = [0.22, 0.9]$, $[\underline{f}^3(\mathbf{x}'), \bar{f}^3(\mathbf{x}')] = [0, 0.27]$, $[\underline{f}^4(\mathbf{x}'), \bar{f}^4(\mathbf{x}')] = [0.055, 0.60]$, and rule consequents are T1 FSs, with $c(G^1) = -2.29$, $c(G^2) = -1.88$, $c(G^3) = 1.88$ and $c(G^4) = 2.29$. What are L and R?

8.14 (Mendel et al. 2014, Chap. 3) Compute $y_l^{\text{COS}}(\mathbf{x}')$ and $y_r^{\text{COS}}(\mathbf{x}')$ when: $M = 4$, $[\underline{f}^1(\mathbf{x}'), \bar{f}^1(\mathbf{x}')] = [0.1, 0.6]$, $[\underline{f}^2(\mathbf{x}'), \bar{f}^2(\mathbf{x}')] = [0.3, 0.8]$, $[\underline{f}^3(\mathbf{x}'), \bar{f}^3(\mathbf{x}')] = [0.1, 0.6]$, $[\underline{f}^4(\mathbf{x}'), \bar{f}^4(\mathbf{x}')] = [0.2, 0.7]$, and rule consequents are IT2 FSs, with

$C(\tilde{G}^1) = [-81.43, -68.57]$, $C(\tilde{G}^2) = [-81.43, -68.57]$, $C(\tilde{G}^3) = [8.57, 21.42]$ and $C(\tilde{G}^4) = [98.57, 111.42]$. What are L and R?

8.15 (Mendel and Wu 2010, Chap. 5, © IEEE 2010) As for the IWA, let $x_i \in [a_i, b_i]$ and $w_i \in [c_i, d_i]$ $(i = 1, \ldots, n)$, but, unlike the IWA, where the membership grade for each x_i and w_i is 1, now the membership grade for each $x_i = x_i'$ and $w_i = w_i'$ is $\mu_{X_i}(x_i')$ and $\mu_{W_i}(w_i')$, respectively. So, now T1 FSs X_i and W_i are associated with $x_i \in [a_i, b_i]$ and $w_i \in [c_i, d_i]$, respectively.
Again the weighted average in (8.1) is evaluated over the Cartesian product space

$$D_{X_1} \times D_{X_2} \times \cdots \times D_{X_n} \times D_{W_1} \times D_{W_2} \times \cdots \times D_{W_n}$$

making use of $\mu_{X_1}(x_1), \mu_{X_2}(x_2), \ldots, \mu_{X_n}(x_n)$ and $\mu_{W_1}(w_1), \mu_{W_2}(w_2), \ldots, \mu_{W_n}(w_n)$, the result being a specific numerical value, y, as well as a degree of membership $\mu_{Y_{\mathrm{FWA}}}(y)$. The result of each pair of computations is the pair $(y, \mu_{Y_{\mathrm{FWA}}}(y))$:

$$\begin{aligned}&\{(x_1, \mu_{X_1}(x_1)), \ldots, (x_n, \mu_{X_n}(x_n)), (w_1, \mu_{W_1}(w_1)), \ldots, (w_n, \mu_{W_n}(w_n))\}\\ &\quad \rightarrow \left(y = \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n {}_i w_i}, \mu_{Y_{\mathrm{FWA}}}(y) \right)\end{aligned}$$

When this is done for all elements in the Cartesian product space, the *fuzzy weighted average* (FWA),Y_{FWA}, is obtained. Observe that Y_{FWA} is itself a T1 FS that is characterized by its MF $\mu_{Y_{\mathrm{FWA}}}(y)$.

Of course, it is impossible to compute $\mu_{Y_{\mathrm{FWA}}}(y)$ for all values of y as just described because to do so would require an uncountable number of computations. Instead, the α-cuts Decomposition Theorem 2.4 is used, as follows:

1. For each $\alpha \in [0, 1]$, the corresponding α-cuts of the T1 FSs X_i and W_i must first be computed, i.e., compute $(i = 1, \ldots, n)$ $X_i \in [a_i(\alpha), b_i(\alpha)]$ and $W_i \in [c_i(\alpha), d_i(\alpha)]$.
2. For each $\alpha \in [0, 1]$, compute the α-cut of the FWA by recognizing that it is an IWA, i.e., $Y_{\mathrm{FWA}}(\alpha) = Y_{\mathrm{IWA}}(\alpha)$, where $Y_{\mathrm{IWA}}(\alpha) = [y_l(\alpha), y_r(\alpha)]$ in which

$$y_l(\alpha) = \min_{\forall w_i(\alpha) \in [c_i(\alpha), d_i(\alpha)]} \frac{\sum_{i=1}^n a_i(\alpha) w_i(\alpha)}{\sum_{i=1}^n w_i(\alpha)} \quad \text{and}$$

$$y_r(\alpha) = \max_{\forall w_i(\alpha) \in [c_i(\alpha), d_i(\alpha)]} \frac{\sum_{i=1}^n b_i(\alpha) w_i(\alpha)}{\sum_{i=1}^n w_i(\alpha)}$$

From (8.15) and (8.16),

$$y_l(\alpha) = \frac{\sum_{i=1}^{L(\alpha)} a_i(\alpha)d_i(\alpha) + \sum_{i=L(\alpha)+1}^{n} a_i(\alpha)c_i(\alpha)}{\sum_{i=1}^{L(\alpha)} d_i(\alpha) + \sum_{i=L(\alpha)+1}^{n} c_i(\alpha)} \quad \text{and}$$
$$y_r(\alpha) = \frac{\sum_{i=1}^{R(\alpha)} b_i(\alpha)c_i(\alpha) + \sum_{i=R(\alpha)+1}^{n} b_i(\alpha)d_i(\alpha)}{\sum_{i=1}^{R(\alpha)} c_i(\alpha) + \sum_{i=R(\alpha)+1}^{n} d_i(\alpha)}$$

where $a_1(\alpha) \le a_2(\alpha) \le \cdots \le a_n(\alpha)$ and $b_1(\alpha) \le b_2(\alpha) \le \cdots \le b_n(\alpha)$. EIASC or EKM algorithms can be used to compute switch points $L(\alpha)$ and $R(\alpha)$. In practice, a finite number of α-cuts are used, so that $\alpha \in [0,1] \to \{\alpha_1, \alpha_2, \ldots, \alpha_{k_{\max}}\}$.

3. Connect all left coordinates $(y_l(\alpha), \alpha)$ and all right coordinates $(y_r(\alpha), \alpha)$ to form the T1 FS Y_{FWA}.

Compute the FWA when: $n = 5$, and X_i and W_i are described by the following triangle MFs, Triangle(e_i, f_i, g_i) :

$$\begin{aligned}
&\mu_{X_1}(x_1) = \text{Triangle}(8.2, 9.0, 9.8), && \mu_{X_2}(x_2) = \text{Triangle}(5.8, 7.0, 8.2),\\
&\mu_{X_3}(x_3) = \text{Triangle}(2.0, 5.0, 8.0), && \mu_{X_4}(x_4) = \text{Triangle}(3.0, 4.0, 5.0),\\
&\mu_{X_5}(x_5) = \text{Triangle}(0.50, 1.0, 1.5), && \mu_{W_1}(w_1) = \text{Triangle}(1.0, 2.0, 3.0),\\
&\mu_{w_2}(w_2) = \text{Triangle}(0.6, 1.0, 1.4), && \mu_{W_3}(w_3) = \text{Triangle}(7.1, 8.0, 8.9),\\
&\mu_{W_4}(w_4) = \text{Triangle}(2.4, 4.0, 5.6), && \mu_{W_5}(w_5) = \text{Triangle}(5.0, 6.0, 7.0)
\end{aligned}$$

8.16 Prove that if the GT2 FS $\tilde{A}$ is totally symmetrical, then $C_{\tilde{A}}(x)$ is symmetrical about $x = m$, and the mean value (i.e., the defuzzified value) of $C_{\tilde{A}}(x)$ equals m. [Hint: Use Fact 1 that is stated on p. 406, about the centroid and defuzzified value of a symmetrical IT2 FS.]

8.17 Complete the proof of the second line of (8.75).

8.18 (a) Prove Properties 8.9–8.11, and (b) provide a figure that is analogous to Fig. 8.16, but for finding $y_r(R)$.

8.19 (a) Derive the $c_l(\tilde{A})$ and $c_r(\tilde{A})$ formulas that are given in (8.87) and (8.88) for the Fig. 8.17 IT2 fuzzy granule; and (b) show that the defuzzified value of the centroid, $x_D = [c_l(\tilde{A}) + c_r(\tilde{A})]/2$, does not depend on the vertical dimension parameters (R and L) of the granule, whereas $c_r(\tilde{A}) - c_l(\tilde{A})$, the centroid length, does depend on R and L, thereby demonstrating that x_D is not a useful measure of the uncertainty of an IT2 fuzzy granule, whereas the centroid is.

References

Alefeld, G. 1981. On the convergence of Halley's method. *American Mathematical Monthly* 88 (7): 530–536.

Almaraashi, M., R. John, A. Hopgood, and S. Ahmadi 2016. Learning of interval and general type-2 fuzzy logic systems using simulated annealing: Theory and practice. *Information Sciences* 360: 21–42.

Chen, C.-L., S.-C. Chen, and Y.-H. Kuo. 2014. The reduction of interval type-2 LR fuzzy sets. *IEEE Transactions on Fuzzy Systems* 22: 840–858.

Cover, T.M., and J.A. Thomas. 1991. *Elements of information theory*. New York: Wiley.

Duran, K., H. Bernal, and M. Melgarejo. 2008. Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set. *NAFIPS 2008*, Paper 50056, New York City, May 2008.

Greenfield, S., R.I. John, and S. Coupland. 2005. A novel sampling method for type-2 defuzzification. In *Proceedings of UKCI 2005*, 120–127, London, September 2005.

Han, S., and X. Liu. 2016. Global convergence of Karnik-Mendel algorithms. *Fuzzy Sets and Systems*.

Hu, H., Y. Wang, and Y. Cai. 2012a. Advantages of the enhanced opposite direction searching algorithm for computing the centroid of an interval type-2 fuzzy set. *Asian Journal of Control* 14 (6): 1–9.

Hu, H., G. Zhao, and H.N. Yang. 2012b. Fast algorithm to calculate generalized centroid of interval type-2 fuzzy set. *Control and Decision* 25 (4): 637–640.

John, R.I. 2000. *Perception modelling using type-2 fuzzy sets*. Ph. D. thesis, De Montfort University.

Karnik, N.N., and J.M. Mendel. 1998. *An introduction to type-2 fuzzy logic systems*. USC-SIPI Report #418, University of Southern California, Los Angeles, CA, June 1998. This can be accessed at: http://sipi.usc.edu/research; then choose sipi technical reports/418.

Karnik, N.N., and J.M. Mendel. 2001. Centroid of a type-2 fuzzy set. *Information Sciences* 132: 195–220.

Karnik, N.N., J.M. Mendel, and Q. Liang. 1999. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 7: 643–658.

Klir, G.J., and T.A. Folger. 1988. *Fuzzy sets, uncertainty, and information*. Englewood Cliffs, NJ: Prentice Hall.

Li, C., J. Yi, and D. Zhao. 2008. A novel type-reduction method for interval type-2 fuzzy logic systems. In: *Proceedings 5th international conference on fuzzy systems knowledge discovery*, vol. 1, 157–161, Jinan, China.

Linda, O., and M. Manic. 2012. Monotone centroid flow algorithm for type reduction of general type-2 fuzzy sets. *IEEE Transactions on Fuzzy Systems* 20: 805–819.

Liu, F. 2008. An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. *Information Sciences* 178: 2224–2236.

Liu, X., Y. Qin, and L. Wu. 2012b. Fast and direct Karnik-Mendel algorithm computation for the centroid of an interval type-2 fuzzy set. In: *Proceedings of FUZZ-IEEE 2012*, 1058–1065, Brisbane, AU, June 2012.

Liu, F., and J.M. Mendel. 2008. Aggregation using the fuzzy weighted average, as computed by the KM algorithms. *IEEE Transactions on Fuzzy Systems* 16: 1–12.

Liu, X., and J.M. Mendel. 2011. Connect Karnik-Mendel algorithms to root-finding for computing the centroid of an interval type-2 fuzzy set. *IEEE Transactions on Fuzzy Systems* 19: 652–665.

Liu, X., J.M. Mendel, and D. Wu. 2012a. Study on enhanced Karnik-Mendel algorithms: Initialization explanations and computation improvements. *Information Sciences* 187: 75–91.

Lucas, L.A., T.M. Centeno, and R.M. Delgado. 2007. General type-2 inference systems: Analysis, design and computational aspects. In *Proceedings of FUZZ-IEEE-2007*, London, UK, 1107–1112.

Melgarejo, M.C.A. 2007. A fast recursive method to compute the generalized centroid of an interval type-2 fuzzy set. In *Proceedings of North American fuzzy information processing society (NAFIPS)*, 190–194.

Mendel, J.M. 2001. *Introduction to rule-based fuzzy logic systems*. Upper Saddle River, NJ: Prentice-Hall.

Mendel, J.M. 2005. On a 50% savings in the computation of the centroid of a *symmetrical* interval type-2 fuzzy set. *Information Sciences* 172: 417–430.

Mendel, J.M. 2007. Advances in type-2 fuzzy sets and systems. *Information Sciences* 177: 84–110.

Mendel, J.M. 2013. On KM algorithms for solving type-2 fuzzy set problems. *IEEE Transactions on Fuzzy Systems* 21: 426–446.

Mendel, J.M. 2014. General type-2 fuzzy logic systems made simple: A tutorial. *IEEE Transactions on Fuzzy Systems* 22: 1162–1182.

Mendel, J.M. 2015. Type-2 fuzzy sets and systems: A retrospective. *Informatik Spektrum* 38 (6): 523–532.

Mendel, J.M., and R.I. John. 2002. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems* 10: 117–127.

Mendel, J.M., and H. Wu. 2006. Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 1, forward problems. *IEEE Transactions on Fuzzy Systems* 14: 781–792.

Mendel, J.M., and H. Wu. 2007. New results about the centroid of an interval type-2 fuzzy set, including the centroid of a fuzzy granule. *Information Sciences* 177: 360–377.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: Aiding people in making subjective judgments*. Hoboken, NJ: Wiley and IEEE Press.

Mendel, J.M., F. Liu, and D. Zhai. 2009. Alpha-plane representation for type-2 fuzzy sets: Theory and applications. *IEEE Transactions on Fuzzy Systems* 17: 1189–1207.

Mendel, J.M., H. Hagras, W.-W. Tan, W.W. Melek, and H. Ying. 2014. *Introduction to type-2 fuzzy logic control*. Hoboken, NJ: Wiley and IEEE Press.

Nie, M., and W.W. Tan. 2008. Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In *Proceedings of IEEE FUZZ conference*, Paper # FS0339, Hong Kong, China, June 2008.

Niewiadomski, A., J. Ochelska, and P.S. Szczepaniak. 2006. Interval-valued linguistic summaries of databases. *Control & Cybernetics* 35 (2): 415–443.

Salaken, S.M., A. Khosravi, and S. Nahavandi. 2016. Modification on enhanced Karnik-Mendel algorithm. *Expert Systems With Applications*.

Starczewski, J.T. 2009. Efficient triangular type-2 fuzzy logic systems. *International Journal of Approximate Reasoning* 50: 799–811.

Tjalling, J.Y. 1995. Historical development of the Newton-Raphson method. *SIAM Review* 37 (4): 531–551.

Ulu, C., M. Güzellkaya, and I. Eksin. 2013. A closed form type reduction method for piecewise linear interval type-2 fuzzy sets. *International Journal of Approximate Reasoning* 54: 1421–1433.

Wu, D. 2011. An interval type-2 fuzzy logic system cannot be implemented by traditional type-1 fuzzy logic systems. In *World conference on soft computing*, San Francisco, CA, May 2011.

Wu, D. 2012. On the fundamental differences between interval type-2 and type-1 fuzzy logic controllers. *IEEE Transactions on Fuzzy Systems* 20: 832–848.

Wu, D. 2013. Approaches for reducing the computational costs of interval type-2 fuzzy logic systems: Overview and comparisons. *IEEE Transactions on Fuzzy Systems* 21 (1): 80–93.

Wu D., and M. Nie. 2011. Comparison and practical implementations of type-reduction algorithms for type-2 fuzzy sets and systems. In *Proceedings of FUZZ-IEEE 2011*, 2131–2138, Taipei, Taiwan, June 2011.

Wu, H., and J.M. Mendel. 2002. Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 10: 622–639.

Wu, D., and J.M. Mendel. 2007a. Aggregation using the linguistic weighted average and interval type-2 fuzzy sets. *IEEE Transactions on Fuzzy Systems* 15 (6): 1145–1161.

Wu, H., and J.M. Mendel. 2007b. Classification of battlefield ground vehicles using acoustic features and fuzzy logic rule-based classifiers. *IEEE Transactions on Fuzzy Systems* 15: 56–72.

Wu, D., and J.M. Mendel. 2009. Enhanced Karnik-Mendel algorithms. *IEEE Transactions on Fuzzy Systems* 17: 923–934.

Wu, H.-J., Y.-L. Su, and S.-J. Lee. 2012. A fast method for computing the centroid of a type-2 fuzzy set. *IEEE Transactions on Systems, Man, and Cybernetics-Part B (Cybernetics)* 42: 764–777.

Xie, B.-K., and S.-J. Lee. 2016. An extended type-reduction method for general type-2 fuzzy sets. *IEEE Transactions on Fuzzy Systems*, accepted for publication, March 2016.

Yeh, C.-Y., W.-H. Roger Jeng, and S.-J. Lee. 2011. An enhanced type-reduction algorithm for type-2 fuzzy sets. *IEEE Transactions on Fuzzy Systems* 19: 227–240.

Zhai, D., and J.M. Mendel. 2011. Computing the centroid of a general type-2 fuzzy set by means of the centroid flow algorithm. *IEEE Transactions on Fuzzy Systems* 19: 401–422.

Zhai, D., and J.M. Mendel. 2012. Enhanced centroid-flow algorithm for computing the centroid of general type-2 fuzzy sets. *IEEE Transactions on Fuzzy Systems* 20: 939–956.

Chapter 9
Interval Type-2 Fuzzy Systems

9.1 Introduction

This chapter is about interval type-2 (IT2) fuzzy systems in which all of the fuzzy sets are interval type-2 fuzzy sets (IT2 FSs), whereas Chap. 11 is about general type-2 (GT2) fuzzy systems. Strictly speaking, only one of the fuzzy sets that are associated with an IT2 fuzzy system has to be an IT2 FS for the resulting fuzzy system to be called an IT2 fuzzy system.

There are now two kinds of IT2 fuzzy systems, the original and still very widely used IT2 fuzzy system that includes type-reduction followed by (+) defuzzification, and the IT2 fuzzy system that uses direct defuzzification. When the first edition of this book was published (Mendel 2001) there was only one kind of IT2 fuzzy system, namely the one with type-reduction + defuzzification. Because type-reduction involves iterative computations, which may introduce undesirable time delays when an IT2 fuzzy system is used in a real-time application such as fuzzy logic control (FLC), an alternative to type-reduction + defuzzification was needed, the result being an IT2 fuzzy system with direct defuzzification. Both kinds of IT2 fuzzy systems (see Fig. 9.1) are described in this chapter, as well as very interesting connections between them.

Observe that the block diagram for the IT2 fuzzy system with direct defuzzification looks exactly like the Fig. 3.1 block diagram for a type-1 fuzzy system. Of course, the difference between them is that in Fig. 9.1b the fuzzy sets are IT2 FSs, whereas in Fig. 3.1 they are T1 FSs.

As in Chap. 3, it is a specific value of $\mathbf{x}$, namely $\mathbf{x}'$, that excites the IT2 fuzzy system, so that the crisp output y depends on it as $f(\mathbf{x}')$. Exactly what this nonlinear function is will be established in this chapter for the two kinds of IT2 fuzzy systems.

Note that the tenets of a fuzzy system do not change from type-1 to type-2 fuzzy sets, and in general, do not change for type-*n*. A higher type number just indicates a higher degree of fuzziness. Since a higher type changes the nature of the

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_9

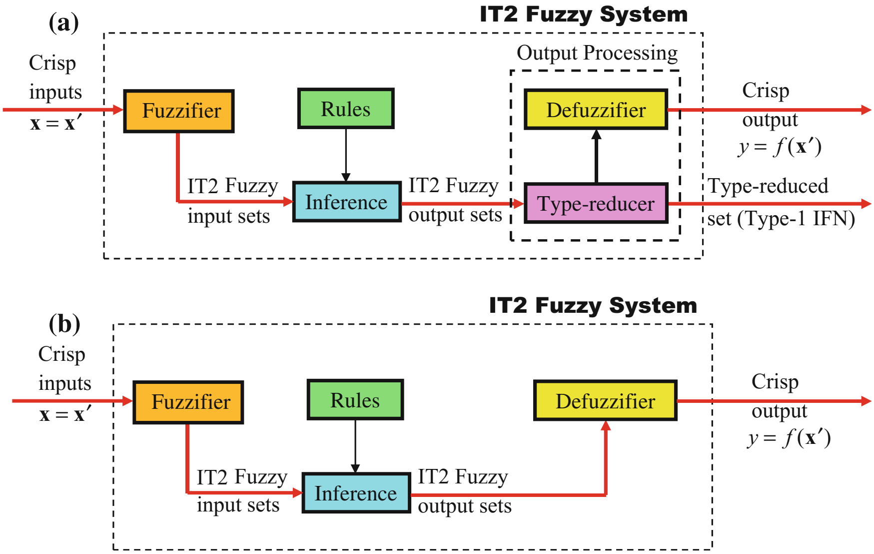


Fig. 9.1 IT2 fuzzy system with: **a** type-reduction + defuzzification and **b** direct defuzzification. T1 IFN is short for "Type-1 interval fuzzy number" (Definition 2.6)

membership functions (MFs), the operations that depend on the MFs change; however, the basic principles of a fuzzy system are independent of the nature of MFs and hence do not change. Rules of inference, like Generalized Modus Ponens, continue to apply.

The structure of this chapter parallels the structure of Chap. 3, wherever possible. In this way, a reader will be able to observe what changes, and how it changes, as one goes from a T1 fuzzy system to an IT2 fuzzy system.

9.2 Rules

Just as the rules of a type-1 fuzzy system can have two different canonical structures, Zadeh and TSK, rules of an IT2 fuzzy system can also have these two different structures. The distinction between type-1 and type-2 is associated with the nature of the MFs, which is not important when forming the rules. Paraphrasing Gertrude Stein, "a rule, is a rule, is a rule." The structure of the rules remains exactly the same in the type-2 case, but now some or all of the sets involved are type-2. However, in order to distinguish between rules that use type-1 or IT2 FSs, the latter will be called *IT2 rules*.

As in a T1 fuzzy system, an IT2 fuzzy system has p inputs $x_1 \in X_1, \ldots, x_p \in X_p$, and one output $y \in Y$, but each x_i is now described by Q_i linguistic terms that are modeled as IT2 FSs $(T_{x_i} = \{\tilde{X}_{ij}\}_{j=1}^{Q_i})$ and y is either described by Q_y linguistic terms that are modeled as IT2 FSs $(T_y = \{\tilde{Y}_j\}_{j=1}^{Q_y})$ or by a function $g(x_1, \ldots, x_p)$.

Definition 9.1 The structure of the lth generic IT2 *Zadeh rule* for an IT2 fuzzy system is $(l = 1, \ldots, M)$:

$$\tilde{R}_Z^l : \text{ IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \tilde{F}_p^l, \text{ THEN } y \text{ is } \tilde{G}^l \tag{9.1}$$

whereas the structure of the lth generic IT2 *Takagi, Sugeno, and Kang* (*TSK*) *rule* for an IT2 fuzzy system is $(l = 1, \ldots, M)$:

$$\tilde{R}_{\text{TSK}}^l : \text{ IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \tilde{F}_p^l, \text{ THEN } y \text{ is } g^l(x_1, \ldots, x_p) \tag{9.2}$$

In both (9.1) and (9.2), $\tilde{F}_1^l \in T_{x_1}, \ldots,$ and $\tilde{F}_p^l \in T_{x_p}$. In (9.1), because $\tilde{G}^l \in T_y$ is an IT2 FS, it is described by its MF $\mu_{\tilde{G}^l}(y, u)$ or FOU. In (9.2), although y does not seem to be a fuzzy set it can be modeled as a *type-2 fuzzy singleton* (see Definition 6.26) $\tilde{G}^l$, so it is made to resemble an IT2 Zadeh rule, where

$$\mu_{\tilde{G}^l(y)}(u) \equiv \begin{cases} 1/1 & \text{when } y = g^l(\mathbf{x}) \\ 1/0 & \text{otherwise} \end{cases} \tag{9.3}$$

In (9.3), $\mathbf{x} = \text{col}(x_1, \ldots, x_p)$.

It is worth noting that GT2 Zadeh and TSK rules (Definition 11.3) have exactly the same formulas as (9.1)–(9.3), except that in Definition 11.3 "IT2" is replaced by "GT2." This is due, of course, to an IT2 FS being a special case of a GT2 FS. Consequently, the rules in Definition 9.1 can also be called T2 Zadeh or TSK rules.

As in Chap. 3, the rules in (9.1) and (9.2) are *complete IF rules* because all p inputs are present in their antecedents. These generic rules represent IT2 fuzzy relations between the input space $X_1 \times \cdots \times X_p$ and the output space, Y, of the IT2 fuzzy system, and which rule structure to use is, as is true for a T1 fuzzy system, very much application dependent. Nonobvious IT2 rules are treated exactly the same as are nonobvious T1 rules, as is explained in Sect. 3.2.

Example 9.1 The IT2 TSK rules that have been used to-date all assume that g^l has a linear dependency upon $x_1, \ldots,$ and x_p. When

$$g^l = c_0^l + c_1^l x_1 + c_2^l x_2 + \cdots + c_p^l x_p \tag{9.4}$$

in which the c_i^l coefficient are numbers, the IT2 TSK fuzzy system that uses these rules is referred to as "A2-C0," meaning that the antecedents are IT2 FSs whereas the consequents are crisp numbers. When

$$G^l = C_0^l + C_1^l x_1 + C_2^l x_2 + \cdots + C_p^l x_p \tag{9.5}$$

in which the C_i^l coefficients are *type-1 interval fuzzy numbers* (see Definition 2.6), the IT2 TSK fuzzy system that uses these rules is referred to as "A2-C1," meaning that the antecedents are IT2 FSs whereas the consequents are T1 FSs.

Definition 9.2 When a fuzzy system uses IT2 Zadeh rules and a Mamdani implication operator it will be referred to as an IT2 *Mamdani fuzzy system*.

Definition 9.3 When a fuzzy system uses IT2 TSK rules and a Mamdani implication operator it will be referred to as an IT2 *TSK fuzzy system*.

Example 9.2 IT2 TSK rules are used in IT2 FLC (more will be said about IT2 FLC in Sect. 10.8). Rules for a discrete-time IT2 TSK fuzzy logic controller often have the following A2-C0 structure $(l = 1, \ldots, M)$:

$$\begin{aligned} \tilde{R}^l_{\text{TSK}}:\ & \text{IF } x(k) \text{ is } \tilde{F}_1^l \text{ and } x(k-1) \text{ is } \tilde{F}_2^l \text{ and } \cdots \text{ and } x(k-p+1) \text{ is } \tilde{F}_p^l, \text{ THEN} \\ & u_l = c_1^l x(k) + c_2^l x(k-1) + \cdots + c_p^l x(k-p+1) \end{aligned} \tag{9.6}$$

In (9.6), $x(k)$ and its time-delayed versions are the states of the dynamical system under control, and the consequent is the control u_l that here is a linear combination of the p states.

Rules for a continuous-time IT2 TSK fuzzy logic controller often have the following A2-C0 structure $(l = 1, \ldots, M)$:

$$\begin{aligned} \tilde{R}^l_{\text{TSK}}:\ & \text{IF } x_1(t) \text{ is } \tilde{F}_1^l \text{ and } x_2(t) \text{ is } \tilde{F}_2^l \text{ and } \cdots \text{ and } x_p(t) \text{ is } \tilde{F}_p^l, \text{ THEN} \\ & u_l = c_1^l x_1(t) + c_2^l x_2(t) + \cdots + c_p^l x_p(t) \end{aligned} \tag{9.7}$$

In (9.7), $x_i(t)$ denotes the ith state of the dynamical system under control, and the consequent is the control u_l that is also a linear combination of the p states.

9.3 Fuzzifier

For a type-1 fuzzy system, two kinds of fuzzifiers were established, singleton and non-singleton. For an IT2 fuzzy system, three kinds of fuzzifiers are possible: singleton, type-1 non-singleton, and IT2 non-singleton.

Definition 9.4 A *type-2 singleton fuzzifier* is one for which[1] $(i = 1, \ldots, p)$ $\mu_{\tilde{X}_i(x_i|x_i')} = 1/1$ when $x_i = x_i'$ and $\mu_{\tilde{X}_i(x_i|x_i')} = 1/0$ when $x_i \neq x_i'$ and $x_i \in X_i$. This will be referred to as a *singleton fuzzifier* when it is used in the context of a type-2 fuzzy system.

Definition 9.5 A *type-1 non-singleton fuzzifier* maps measurement $(i = 1, \ldots, p)$ $x_i = x_i'$ into a type-1 fuzzy number (see Definition 2.5) for which $\mu_{X_i}(x_i') = 1$ and $\mu_{X_i}(x_i)$ decreases from unity as x_i moves away from x_i'. Because the MF for x_i' will be that of a type-1 fuzzy number it is expressed as $\mu_{X_i}(x_i|x_i')$. When used in an IT2 fuzzy system, this fuzzifier is expressed as (see Sect. 6.8) $1/\mu_{X_i}(x_i|x_i')$.

Definition 9.6 An *IT2 non-singleton fuzzifier* maps measurement $x_i = x_i'$ into an IT2 fuzzy number (Definition 6.22), i.e., it is an IT2 FS whose lower and upper MFs of its FOU are type-1 fuzzy numbers.

A popular IT2 fuzzy number is the Gaussian primary MF with uncertain standard deviation (see Example 6.16), whose mean is located at x_i'.

A situation where an IT2 non-singleton fuzzifier is of great value is in time-series forecasting when the additive measurement noise is *nonstationary*. This can be interpreted as noise whose probability density function changes with time. The FOU associated with the IT2 non-singleton fuzzifier provides a natural way within the framework of fuzzy sets to model such nonstationarity.

Sahab and Hagras (2011, 2012) take a different approach to IT2 non-singleton fuzzification. They state

> … all papers (about type-2 non-singleton T2) assume a predefined shape (such as triangular or Gaussian) as the fuzzy input variable. However, these predefined shapes might not accurately represent the real shape of the data distribution related to the input device or a sensor.

Instead of using an IT2 fuzzy number as the model for an IT2 non-singleton fuzzifier, they use an experimental method to determine the FOU for each input variable directly from sensor data. If one has access to such data or can acquire it, then this is an interesting approach to IT2 non-singleton fuzzification. It leads to an IT2 FS fuzzifier whose LMF and UMF are determined from data. Because such a fuzzifier is not parsimonious, it is not used in the rest of this book.

9.4 Fuzzy Inference Engine

This section first obtains results that are not dependent upon the nature of the fuzzifier and are valid for any kind of type-2 fuzzy system. It then specializes, those results to IT2 fuzzy systems for singleton and both kinds of non-singleton fuzzifiers. The more general results are used later in Chap. 11 for GT2 fuzzy systems.

[1]Separable MFs are assumed, as is discussed in Sect. 6.10.

9.4.1 General Results

In a type-1 fuzzy system, the inference engine maps input T1 FSs into fired-rule output T1 FSs. Multiple antecedents in rules are connected by the t-norm. The membership grades in the input sets are combined with those in the output sets using the sup-star composition. Multiple rules may be combined using the t-conorm operation or during defuzzification by weighted summation. In the type-2 case the inference process is very similar. The inference engine maps input T2 FSs into output T2 FSs. To do this, one needs to compute unions and intersections of T2 FSs (using results from Sect. 7.3 for IT2 FSs or Sect. 7.4 for GT2 FSs), as well as compositions of T2 FSs with type-2 relations (using results from Sect. 7.9).

Just as the sup-star composition is the backbone computation for a type-1 fuzzy system, the extended sup-star composition is the backbone computation for a type-2 fuzzy system.

To begin, the focus here is on the T2 Zadeh rules in (9.1), and a T2 Mamdani fuzzy system. Let $\tilde{F}_1^l \times \cdots \times \tilde{F}_p^l = \tilde{A}^l$; then, (9.1) can be re-expressed as $(l = 1, \ldots, M)$:

$$\tilde{R}_Z^l : \tilde{F}_1^l \times \cdots \times \tilde{F}_p^l \to \tilde{G}^l = \tilde{A}^l \to \tilde{G}^l \tag{9.8}$$

$\tilde{R}_Z^l$ is described by the MF $\mu_{\tilde{R}_Z^l}(\mathbf{x}, y)$, where

$$\mu_{\tilde{R}_Z^l}(\mathbf{x}, y) = \mu_{\tilde{A}^l \to \tilde{G}^l}(\mathbf{x}, y) \tag{9.9}$$

It is known, from (7.115), that $(x_1 \in X_1, \ldots, x_p \in X_p)$

$$\begin{aligned} \mu_{\tilde{R}_Z^l}(\mathbf{x}, y) = \mu_{\tilde{A}^l \to \tilde{G}^l}(\mathbf{x}, y) &= \mu_{\tilde{F}_1^l(x_1)} \sqcap \cdots \sqcap \mu_{\tilde{F}_p^l(x_p)} \sqcap \mu_{\tilde{G}^l(y)} \\ &= \left[\sqcap_{i=1}^p \mu_{\tilde{F}_i^l(x_i)} \right] \sqcap \mu_{\tilde{G}^l(y)} \end{aligned} \tag{9.10}$$

Additionally, the p-dimensional input to $\tilde{R}_Z^l$ is given by the T2 FS set $\tilde{A}_{\mathbf{x}'}$ whose MF is [see (6.69)]

$$\mu_{\tilde{A}_{\mathbf{x}'}}(\mathbf{x}) = \mu_{\tilde{X}_1(x_1|x_1')} \sqcap \ldots \sqcap \mu_{\tilde{X}_p(x_p|x_p')} = \sqcap_{i=1}^p \mu_{\tilde{X}_i(x_i|x_i')} \tag{9.11}$$

where $\tilde{X}_i$ $(i = 1, \ldots, p)$ are the labels of the fuzzy sets describing the inputs and $x_1 \in X_1, \ldots, x_p \in X_p$. Their MFs change as the input to the IT2 fuzzy system changes, which is why the conditioning notation is used.

Each Zadeh rule $\tilde{R}_Z^l$ determines a T2 FS $\tilde{B}^l = \tilde{A}_{\mathbf{x}'} \cdot \tilde{R}_Z^l$ $(l = 1, \ldots, M)$ such that [see (7.92)][2]:

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = \mu_{\tilde{A}_{\mathbf{x}'} \cdot \tilde{R}_Z^l}(y|\mathbf{x}') = \sqcup_{\mathbf{x} \in \mathbf{X}} \left[\mu_{\tilde{A}_{\mathbf{x}'}}(\mathbf{x}) \sqcap \mu_{\tilde{R}_Z^l}(\mathbf{x}, y) \right], \quad y \in Y \tag{9.12}$$

This equation is the input–output relation in Fig. 9.1a, b between the T2 FS that excites one rule in the inference block (engine) and the T2 FS at the output of that engine.

Substituting (9.10) and (9.11) into (9.12), it follows that $(l = 1, \ldots, M)$:

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = \sqcup_{\mathbf{x} \in \mathbf{X}} \left\{ \left(\sqcap_{i=1}^p \mu_{\tilde{X}_i(x_i|x_i')} \right) \sqcap \left(\left[\sqcap_{i=1}^p \mu_{\tilde{F}_i^l(x_i)} \right] \sqcap \mu_{\tilde{G}^l(y)} \right) \right\} \quad y \in Y \tag{9.13}$$

which can be expressed as[3]

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = \left\{ \sqcup_{\mathbf{x} \in \mathbf{X}} \left[\left(\sqcap_{i=1}^p \left(\mu_{\tilde{X}_i(x_i|x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)} \right) \right) \right] \right\} \sqcap \mu_{\tilde{G}^l(y)} \quad y \in Y \tag{9.14}$$

This equation is analogous to the second line of (3.11), where $\sqcup$ plays the role of "sup" and $\sqcap$ plays the role of $\bigstar$.

Expanding the large inner bracket in (9.14), that equation can be rewritten as

$$\begin{aligned} \mu_{\tilde{B}^l(y|\mathbf{x}')} = \sqcup_{\mathbf{x} \in \mathbf{X}} & \left[\left(\mu_{\tilde{X}_1(x_1|x')} \sqcap \mu_{\tilde{F}_1^l(x_1)} \right) \sqcap \left(\mu_{\tilde{X}_2(x_2|x_2')} \sqcap \mu_{\tilde{F}_2^l(x_2)} \right) \cdots \sqcap \left(\mu_{\tilde{X}_p(x_p|x_p')} \sqcap \mu_{\tilde{F}_p^l(x_p)} \right) \right] \\ & \sqcap \mu_{\tilde{G}^l(y)} \quad y \in Y \end{aligned} \tag{9.15}$$

Because set theoretic operations are defined only over the *same* universe of discourse, (9.15) becomes

$$\begin{aligned} \mu_{\tilde{B}^l(y|\mathbf{x}')} = & \left\{ \left[\sqcup_{x_1 \in X_1} \left(\mu_{\tilde{X}_1(x_1|x_1')} \sqcap \mu_{\tilde{F}_1^l(x_1)} \right) \right] \sqcap \cdots \sqcap \left[\sqcup_{x_p \in X_p} \left(\mu_{\tilde{X}_p(x_p|x_p')} \sqcap \mu_{\tilde{F}_p^l(x_p)} \right) \right] \right\} \\ & \sqcap \mu_{\tilde{G}^l(y)} \quad y \in Y \end{aligned} \tag{9.16}$$

---

[2]Equation (9.12) is the MF of a vertical slice, which is a T1 FS, whose name is $\tilde{B}^l(y|\mathbf{x}')$, hence, the notation $\mu_{\tilde{B}^l(y|\mathbf{x}')}$.

[3]This uses the Associative and Commutative Laws for the meet, which is okay to do for both the minimum and product t-norms and any kind of secondary MFs (see Tables 7.1 and 7.2).

This equation is analogous to the last line of (3.11), where $\sqcap$ has replaced $\star$ and $\sqcup$ has replaced sup, and it can be expressed more succinctly, as

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = \left\{ \sqcap_{i=1}^{p} \left[\sqcup_{x_i \in X_i} \left(\mu_{\tilde{X}_i(x_i|x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)} \right) \right] \right\} \sqcap \mu_{\tilde{G}^l(y)} \quad y \in Y \tag{9.17}$$

There is a lot going on in (9.17), just as there is a lot going on in (3.11). Given $\mathbf{x} = \mathbf{x}'$, in (9.17) one must compute:

1. $\mu_{\tilde{X}_i(x_i|x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)}$ for $x_i \in X_i$ and $i = 1, \ldots, p$. Each meet is a T1 FS. If, for example, each X_i is discretized into 10 samples, then 10 meets are computed for each of the $p\, x_i$.
2. $\sqcup_{x_i \in X_i} \left(\mu_{\tilde{X}_i(x_i|x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)} \right)$ for $i = 1, \ldots, p$. Each of the p joins is a T1 FS and if, for example, each X_i is discretized into 10 samples, it will be the join of the 10 T1 FS meets computed in Step 1 (for each of the $p\, x_i$).
3. $\sqcap_{i=1}^{p} \left[\sqcup_{x_i \in X_i} \left(\mu_{\tilde{X}_i(x_i|x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)} \right) \right]$. This is a meet of the p join T1 FSs computed in Step 2, and results in one T1 FS.
4. $\left\{ \sqcap_{i=1}^{p} \left[\sqcup_{x_i \in X_i} \left(\mu_{\tilde{X}_i(x_i|x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)} \right) \right] \right\} \sqcap \mu_{\tilde{G}^l(y)}$, $y \in Y$. At each y, this is a T1 FS. If Y is also discretized into 10 samples ($y = y_1, \ldots, y_{10}$), then 10 meets are computed between the one T1 FS from Step 3 and $\mu_{\tilde{G}^l(y_i)}$ ($i = 1, \ldots, 10$).

These are all vertical-slice computations, the result being a vertical-slice representation of $\tilde{B}^l$. For GT2 FSs, the meets and joins can be computed by using Theorems 7.6 and 7.3, respectively (subject to the constraints of those theorems). For IT2 FSs, the meets and joins can be computed by using Theorems 7.12 and 7.11, respectively.

Before Step 1 can be performed one must decide on which fuzzifier to use, because $\mu_{\tilde{X}_i(x_i|x_i')}$ depends on that choice. In Mendel (2001), there is a separate chapter for singleton, type-1, and IT2 fuzzification for IT2 fuzzy systems, which led to a lot of repetition across these three chapters. In this book, as in Chap. 3, the different kinds of fuzzifiers are treated in one chapter, this chapter.

A close examination of (9.17) reveals that terms within its brace involve interactions between a fuzzified input and its respective antecedent. It is only after all of those interactions have been computed that the meet is taken between the braced term and the entire consequent MF, one vertical slice at a time. In this book, the braced term is denoted $F^l(\mathbf{x}')$, where the T1 FS $F^l(\mathbf{x}')$ is called a *firing set*, i.e.,

$$F^l(\mathbf{x}') \equiv \sqcap_{i=1}^{p} \left[\sqcup_{x_i \in X_i} \left(\mu_{\tilde{X}_i(x_i|x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)} \right) \right] \tag{9.18}$$

It will be shown below that when all T2 FSs are IT2 FSs then $F^l(\mathbf{x}')$ is a type-1 interval fuzzy number, whose support is called a *firing interval*. (9.18) is analogous to (3.12).

Examining (9.18), observe that the interaction of each input with its antecedent contributes towards the rule's firing set. Denoting each input-antecedent interaction as $F^l(x'_i)$, where

$$F^l(x'_i) \equiv \sqcup_{x_i \in X_i}\left(\mu_{\tilde{X}_i(x_i|x'_i)} \sqcap \mu_{\tilde{F}^l_i(x_i)}\right) \tag{9.19}$$

it follows, from (9.18) and (9.19), that

$$F^l(\mathbf{x}') = \sqcap_{i=1}^{p} F^l(x'_i) \tag{9.20}$$

Consequently, (9.17) can be expressed very simply, as $(l = 1, \ldots, M)$:

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = F^l(\mathbf{x}') \sqcap \mu_{\tilde{G}^l(y)} \quad y \in Y \tag{9.21}$$

Be sure to compare (9.21) with (3.15) to see their similarity. In words, (9.21) says that the MF for a fired rule is the meet of the rule's firing set and the *entire* consequent fuzzy set. Observe that, for a T2 Mamdani fuzzy system, $\mu_{\tilde{B}^l(y|\mathbf{x}')}$, the MF of the fuzzy set $\tilde{B}^l$, is a function of y, where $y \in Y$.

Focusing next on the TSK rules in (9.2) and a T2 TSK fuzzy system, following the same chain of reasoning used to obtain (9.20), and using (9.3) along with the fact [see (7.123)] $F^l(\mathbf{x}') \sqcap 1/1 = F^l(\mathbf{x}')$, it should be obvious that, for TSK rules, (9.21) becomes $(l = 1, \ldots, M)$:

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = \mu_{\tilde{B}^l(\mathbf{x}')} = F^l(\mathbf{x}') \text{ when } y = g^l(\mathbf{x}') \tag{9.22}$$

Observe that, for a T2 TSK fuzzy system, $\mu_{\tilde{B}^l(\mathbf{x}')}$ is an explicit function of $\mathbf{x}'$ and an implicit function of y, where $\mathbf{x}' \in X_1 \times X_2 \times \cdots \times X_p$. For those who are wondering at this point where or when the consequent $g^l(\mathbf{x}')$ will appear in the T2 TSK fuzzy system, it will appear during type-reduction + defuzzification (see Sect. 9.6.4).

To summarize:

Theorem 9.1 *Let* $\mathbf{x}'$ *denote the input to a T2 fuzzy system. Then the MF for a* fired rule output set, $\tilde{B}^l$, *is* $(l = 1, \ldots, M)$:

$$\begin{cases} \text{T2 Mamdani fuzzy system:} & \mu_{\tilde{B}^l(y|\mathbf{x}')} = F^l(\mathbf{x}') \sqcap \mu_{\tilde{G}^l(y)} \quad y \in Y \\ \text{T2 TSK fuzzy system:} & \mu_{\tilde{B}^l(\mathbf{x}')} = F^l(\mathbf{x}') \text{ when } y = g^l(\mathbf{x}') \end{cases} \tag{9.23}$$

In (9.23), $F^l(\mathbf{x}')$ *is given by (9.18) [or (9.20) and (9.19)].*

This theorem is the type-2 counterpart to Theorem 3.1.

The rest of this chapter is devoted to IT2 Mamdani and TSK fuzzy systems and so (9.23) will need to be examined for such fuzzy systems. It will be examined further for GT2 fuzzy systems in Chap. 11.

9.4.2 Fuzzification and Its Effects on Inference for IT2 Fuzzy Systems

In order to compute the firing set in (9.20), one needs to compute $F^l(x_i)$ in (9.19) $(i = 1, \ldots, p)$. This requires choosing a fuzzifier. In this section $F^l(x_i)$ is evaluated first for a singleton fuzzifier and then for the two kinds of non-singleton fuzzifiers.

9.4.2.1 Singleton Fuzzifier

For (type-2) singleton fuzzification (Definition 9.4) the join and meet operations in (9.23) are easy to evaluate because $\mu_{X_i}(x_i|x_i')$ is nonzero only at one point, $x_i = x_i'$. The result is a corollary to Theorem 9.1.

Corollary 9.1 *For* singleton fuzzification (Liang and Mendel 2000a):

$$\begin{cases} \text{IT2 Mamdani fuzzy system:} & \mu_{\tilde{B}^l(y|\mathbf{x}')} = \left(1/[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]\right) \sqcap \mu_{\tilde{G}^l(y)} \quad y \in Y \\ \text{IT2 TSK fuzzy system:} & \mu_{\tilde{B}^l(\mathbf{x}')} = 1/[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')] \text{ when } y = g^l(\mathbf{x}') \end{cases} \tag{9.24}$$

where

$$\underline{f}^l(\mathbf{x}') = T_{i=1}^p \underline{\mu}_{\tilde{F}_i^l}(x_i') \tag{9.25}$$

$$\bar{f}^l(\mathbf{x}') = T_{i=1}^p \bar{\mu}_{\tilde{F}_i^l}(x_i') \tag{9.26}$$

Proof For singleton fuzzification $\mu_{\tilde{X}_i(x_i|x_i')}$ is nonzero only when $x_i = x_i'$, so that $F^l(\mathbf{x}')$ in (9.20) simplifies as follows:

$$F^l(\mathbf{x}') \equiv \sqcap_{i=1}^p \left[\sqcup_{x_i \in X_i} \left(\mu_{\tilde{X}_i(x_i|x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)}\right)\right] = \sqcap_{i=1}^p \left[(1/1) \sqcap \mu_{\tilde{F}_i^l(x_i')}\right] = \sqcap_{i=1}^p \mu_{\tilde{F}_i^l(x_i')} \tag{9.27}$$

Note that $(1/1) \sqcap \mu_{\tilde{F}_i^l(x_i')} = \mu_{\tilde{F}_i^l(x_i')}$ is a direct application of (7.123). Recall that $\mu_{\tilde{F}_i^l(x_i')}$ is the secondary MF of the IT2 FS $\tilde{F}_i^l(x_i)$ when $x_i = x_i'$, i.e.,

$$\mu_{\tilde{F}_i^l(x_i')} = 1/[\underline{\mu}_{\tilde{F}_i^l}(x_i'), \bar{\mu}_{\tilde{F}_i^l}(x_i')] \tag{9.28}$$

Applying Theorem 7.12 to the right-hand side of (9.27), in which

$$[l_i, r_i] = [\underline{\mu}_{\tilde{F}_i^l}(x_i'),\ \bar{\mu}_{\tilde{F}_i^l}(x_i')] \tag{9.29}$$

it follows that

$$F^l(\mathbf{x}') = 1/[\underline{\mu}_{\tilde{F}_1^l}(x_1')\bigstar\cdots\bigstar\underline{\mu}_{\tilde{F}_p^l}(x_p'),\ \bar{\mu}_{\tilde{F}_1^l}(x_1')\bigstar\cdots\bigstar\bar{\mu}_{\tilde{F}_p^l}(x_p')] \equiv 1/[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')] \tag{9.30}$$

where $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ are given in (9.25) and (9.26), respectively. Substituting (9.30) into (9.23), one obtains (9.24), which competes the proof of Corollary 9.1.

Definition 9.7 In an IT2 fuzzy system $[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]$ is called the *firing interval* at $\mathbf{x} = \mathbf{x}'$. Formulas for the firing interval depend on the kind of fuzzifier that is used in the IT2 fuzzy system. For singleton fuzzification, $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ are given in (9.25) and (9.26), respectively. Observe, from these two equations, that $\underline{f}^l(\mathbf{x}')$ makes use of LMFs of the antecedent FOUs, and $\bar{f}^l(\mathbf{x}')$ makes use of UMFs of the antecedent FOUs. Taken together, however, the firing interval makes use of both the LMFs and UMFs of the antecedent FOUs.

Corollary 9.2 *For* singleton fuzzification *and an* IT2 Mamdani fuzzy system (Liang and Mendel 2000a), *the MF for a* fired-rule output set, $\tilde{B}^l$, *is* $(l = 1, \ldots, M$ and $y \in Y)$:

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = 1/[\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}'),\ \bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}')] = 1/\mathrm{FOU}(\tilde{B}^l(y|\mathbf{x}')) \tag{9.31}$$

where

$$\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}') = \underline{f}^l(\mathbf{x}')\bigstar\underline{\mu}_{\tilde{G}^l}(y) \tag{9.32}$$

$$\bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}') = \bar{f}^l(\mathbf{x}')\bigstar\bar{\mu}_{\tilde{G}^l}(y) \tag{9.33}$$

in which $\underline{f}^l(\mathbf{x}')$ *and* $\bar{f}^l(\mathbf{x}')$ *are given in (9.25) and (9.26), respectively.*

Proof This corollary to Theorem 9.1 is a continuation of Corollary 9.1 in which the top line of (9.24) is completed. The IT2 FS consequent $\tilde{G}^l$ has MF (in vertical-slice notation)

$$\mu_{\tilde{G}^l(y)} = 1/[\underline{\mu}_{\tilde{G}^l}(y), \bar{\mu}_{\tilde{G}^l}(y)] \quad y \in Y; \tag{9.34}$$

hence,

$$\begin{aligned} \mu_{\tilde{B}^l(y|\mathbf{x}')} &= \left(1/[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]\right) \sqcap \mu_{\tilde{G}^l(y)} = \left(1/[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]\right) \\ &\sqcap \left(1/[\underline{\mu}_{\tilde{G}^l}(y), \bar{\mu}_{\tilde{G}^l}(y)]\right) \; y \in Y \end{aligned} \tag{9.35}$$

Applying Theorem 7.12 to (9.35), it follows that

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = 1/\left[\underline{f}^l(\mathbf{x}')\bigstar\underline{\mu}_{\tilde{G}^l}(y), \bar{f}^l(\mathbf{x}')\bigstar\bar{\mu}_{\tilde{G}^l}(y)\right] \quad y \in Y \tag{9.36}$$

Expressing $\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}')$ as in (9.31), it then follows from (9.36) that $\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}')$ are given by (9.32) and (9.33), respectively, which completes the proof of this corollary.

Example 9.3 Here pictorial descriptions of Corollaries 9.1 and 9.2 are obtained for an IT2 Mamdani fuzzy system that uses either minimum or product t-norms, so as to better understand some of the flow of rule-uncertainties through an IT2 fuzzy system.

Figure 9.2 depicts input, antecedent, and firing interval computations for a two-antecedent–single consequent rule, singleton fuzzification, and minimum or product t-norms. In both cases, the firing strength is an interval $[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]$, where $\underline{f}^l(\mathbf{x}') = \underline{\mu}_{\tilde{F}_1^l}(x_1')\bigstar\underline{\mu}_{\tilde{F}_2^l}(x_2')$ and $\bar{f}^l(\mathbf{x}') = \bar{\mu}_{\tilde{F}_1^l}(x_1')\bigstar\bar{\mu}_{\tilde{F}_2^l}(x_2')$, and $\bigstar$ is minimum

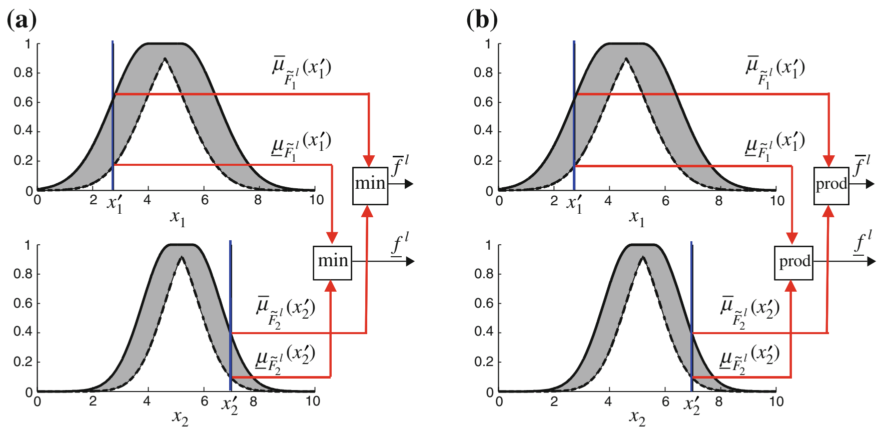


Fig. 9.2 Pictorial description of input, antecedent operations, and firing interval operations for an IT2 fuzzy system. Singleton fuzzification with: **a** minimum t-norm and **b** product t-norm (Liang and Mendel 2000a, p. 540; © 2000 IEEE)

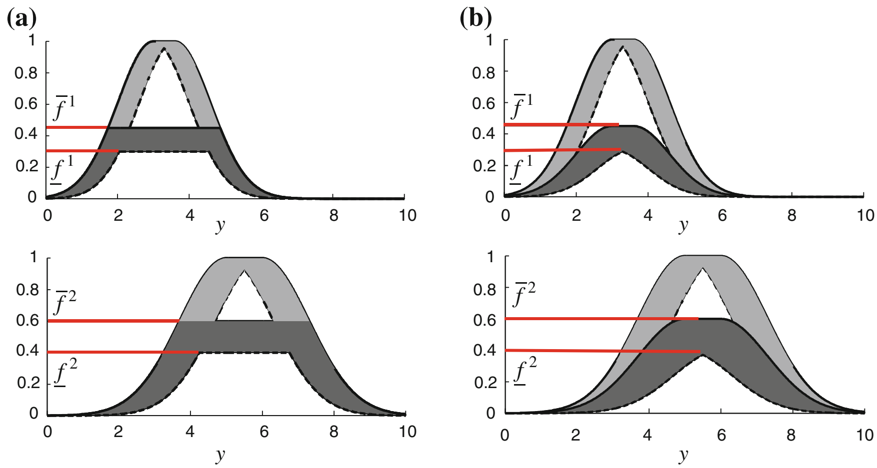


Fig. 9.3 Pictorial description of consequent operations for a two-rule IT2 fuzzy system. Fired-rule output sets with: **a** minimum t-norm, and **b** product t-norm

or product. Observe, for example, that $\underline{\mu}_{\tilde{F}_1^l}(x_1')$ and $\bar{\mu}_{\tilde{F}_1^l}(x_1')$ occur at the intersections of the vertical line at x_1' with $\underline{\mu}_{\tilde{F}_1^l}(x_1)$ and $\bar{\mu}_{\tilde{F}_1^l}(x_1)$, respectively The main thing to observe from Fig. 9.2a, b is that, regardless of the t-norm, the result of input and antecedent operations is an interval—the firing interval.

Figure 9.3 depicts FOU$(\tilde{B}^l)$ for a two-rule (l = 1, 2) singleton IT2 fuzzy system. It is obtained by implementing (9.32) and (9.33) in Corollary 9.2 at each $y \in Y$. For example, $\bar{f}^1(\mathbf{x}')$ is t-normed with the UMF $\bar{\mu}_{\tilde{G}^1}(y)$ to give the upper solid curve $\bar{f}^1(\mathbf{x}')\bigstar\bar{\mu}_{\tilde{G}^1}(y)$ for $y \in Y$, and $\underline{f}^1(\mathbf{x}')$ is t-normed with the LMF $\underline{\mu}_{\tilde{G}^1}(y)$ to give the lower dashed curve $\underline{f}^1(\mathbf{x}')\bigstar\underline{\mu}_{\tilde{G}^1}(y)$ for $y \in Y$. FOU$(\tilde{B}^l)$ is the area between these two functions, which has been darkened.

Example 9.4[4] Here is a totally different way to obtain the results that are given in Corollaries 9.1 and 9.2 that does not need the concepts of join and meet and only uses type-1 mathematics (adapted from Mendel et al. 2006, pp. 815–816). In order to see the forest from the trees, so-to-speak, the focus here is on a single rule (i.e., $l = 1$) that has one antecedent.

In the rule[5] "IF x_1 is $\tilde{F}_1$, THEN y is $\tilde{G}$," let $\tilde{F}_1$ be an IT2 FS in the discrete universe of discourse X_{1d} for the antecedent, and $\tilde{G}$ be an IT2 FS in the discrete

[4]This example can be skipped without any loss of continuity, because it only provides a different derivation of the already-derived results that are in Corollaries 9.1 and 9.2.

[5]Although it is unnecessary to use the subscript 1 on x for a single-antecedent rule, by doing so the multiple-antecedent case in Mendel et al. (2006) will be easier to understand because of the presence of the subscript 1 in all of the notation and formulas.

universe of discourse Y_d for the consequent. Decompose $\tilde{F}_1$ into n_{F_1} embedded IT2 FSs $\tilde{F}_{1e}^{j_1}$ ($j_1 = 1, \ldots, n_{F_1}$), whose domains are the embedded T1 FSs $F_{1e}^{j_1}$, and decompose $\tilde{G}$ into n_G embedded IT2 FSs $\tilde{G}_e^j$ ($j = 1, \ldots, n_G$), whose domains are the embedded T1 FSs G_e^j. According to (6.18), (6.44) and (6.14), $\tilde{F}_1$ and $\tilde{G}$ can be expressed as:

$$\tilde{F}_1 = 1/\mathrm{FOU}(\tilde{F}_1) = 1/\sum_{j_1=1}^{n_{F_1}} F_{1e}^{j_1} \tag{9.37}$$

$$\sum_{j_1=1}^{n_{F_1}} F_{1e}^{j_1} = \sum_{j_1=1}^{n_{F_1}} \sum_{i=1}^{N_{x_1}} u_{1i}^{j_1}/x_{1i}, \quad u_{1i}^{j_1} \in [0,\ 1], \tag{9.38}$$

$$\tilde{G} = 1/\mathrm{FOU}(\tilde{G}) = 1/\sum_{j=1}^{n_G} G_e^j \tag{9.39}$$

$$\sum_{j=1}^{n_G} G_e^j = \sum_{j=1}^{n_G} \sum_{k=1}^{N_y} w_k^j/y_k, \quad w_k^j \in [0,\ 1] \tag{9.40}$$

Consequently, there are $n_{F_1} \times n_G$ possible combinations of embedded type-1 antecedent and consequent FSs so that the totality of fired-rule output sets for all possible combinations of these embedded type-1 antecedent and consequent FSs is a *bundle of functions* $\mathrm{FOU}(\tilde{B}(y|x'))$ as depicted in Fig. 9.4, where $(y \in Y_d)$:

$$\mathrm{FOU}(\tilde{B}(y|x_1')) \triangleq \sum_{j_1=1}^{n_{F_1}} \sum_{j=1}^{n_G} \mu_{B(j_1,j)}(y|x_1') \tag{9.41}$$

in which the summations denote union. Because the secondary grades of $\tilde{F}_1$ and $\tilde{G}$ are unity over their entire FOUs, a secondary grade of unity can also be assigned to FOU $(\tilde{B}\,(y|x_1'))$, so that $(y \in Y_d)$ $\mu_{\tilde{B}(y|x_1')} = 1/\mathrm{FOU}(\tilde{B}\,(y|x_1'))$.

From Fig. 9.4, one can deduce that the fired-rule output of the combination of the j_1th embedded type-1 antecedent FS and the jth embedded type-1 consequent FS can be computed by using the top line of (3.20) with p = 1, i.e.,[6] $(y \in Y_d)$

$$\mu_{B(j_1,j)}(y|x_1') = \mu_{F_{1e}^{j_1}}(x_1') \bigstar \mu_{G_e^j}(y) \tag{9.42}$$

[6]In (3.20), the superscript l denotes rule number. Since the focus here is on a single rule, this superscript is not used here. Instead, superscripts are associated with specific embedded type-1 FSs.

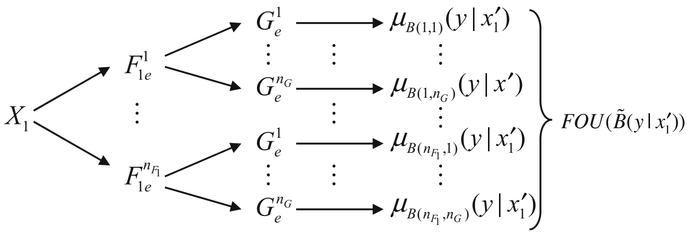


Fig. 9.4 Fired-rule output FSs for all possible $n_B = n_{F_1} \times n_G$ combinations of the embedded type-1 antecedent and consequent FSs for a single-antecedent rule (Mendel et al. 2006, © 2006, IEEE)

Since for any j_1 and j, $\mu_{B(j_1,j)}(y|x_1')$ in (9.42) is bounded in [0, 1], FOU$(\tilde{B}\,(y|x_1'))$ in (9.41) must also be a bounded function in [0, 1], which means that (9.41) can be expressed as a set of $n_{F_1} \times n_G$ functions $(y \in Y_d)$:

$$\text{FOU }(\tilde{B}(y|x_1')) \equiv \left\{\underline{\mu}_{\tilde{B}}(y|x_1'), \ldots, \bar{\mu}_{\tilde{B}}(y|x_1')\right\} \tag{9.43}$$

$$\underline{\mu}_{\tilde{B}}(y|x_1') = \inf_{\forall j_1,j}(\mu_{B(j_1,j)}(y|x_1')) \tag{9.44}$$

$$\bar{\mu}_{\tilde{B}}(y|x_1') = \sup_{\forall j_1,j}(\mu_{B(j_1,j)}(y|x_1')) \tag{9.45}$$

Let $\bar{\mu}_{\tilde{F}_1}(x_1)$ and $\underline{\mu}_{\tilde{F}_1}(x_1)$ denote the upper and lower MFs for FOU$(\tilde{F}_1)$, and $\bar{\mu}_{\tilde{G}}(y)$ and $\underline{\mu}_{\tilde{G}}(y)$ denote the upper and lower MFs for FOU$(\tilde{G})$. From (9.42), observe that to compute the *infimum* of $\mu_{B(j_1,j)}(y|x_1')$ one needs to choose the smallest embedded type-1 FS of both the antecedent and consequent, namely $\underline{\mu}_{\tilde{F}_1}(x_1)$ and $\underline{\mu}_{\tilde{G}}(y)$, respectively. By doing this, the following equation is obtained for $\underline{\mu}_{\tilde{B}}(y|x_1')$ $(y \in Y_d)$:

$$\underline{\mu}_{\tilde{B}}(y|x_1') = \inf_{\forall j_1,j}(\mu_{B(j_1,j)}(y|x_1')) = \underline{\mu}_{\tilde{F}_1}(x_1')\bigstar\underline{\mu}_{\tilde{G}}(y) \tag{9.46}$$

Similarly, to compute the *supremum* of $\mu_{B(j_1,j)}(y|x_1')$, one needs to choose the largest embedded type-1 FS of both the antecedent and consequent, namely $\bar{\mu}_{\tilde{F}_1}(x_1)$ and $\bar{\mu}_{\tilde{G}}(y)$, respectively. By doing this, the following equation is obtained for $\bar{\mu}_{\tilde{B}}(y|x_1')$ $(y \in Y_d)$:

$$\bar{\mu}_{\tilde{B}}(y|x') = \sup_{\forall j_1,j}(\mu_{B(j_1,j)}(y|x')) = \bar{\mu}_{\tilde{F}_1}(x_1')\bigstar\bar{\mu}_{\tilde{G}}(y) \tag{9.47}$$

Obviously, when the sample rate becomes infinite, the sampled universes of discourse X_{1d} and Y_d can be considered as the continuous universes of discourse X_1 and Y, respectively. In this case, $\mathrm{FOU}(\tilde{B}\,(y|x'))$ contains an infinite and uncountable number of elements, which will still be bounded below and above by $\underline{\mu}_{\tilde{B}}(y|x_1')$ and $\bar{\mu}_{\tilde{B}}(y|x_1')$, respectively, where these functions are still given by (9.46) and (9.47) (with $Y_d \to Y$), such that (9.43) can be expressed as $(y \in Y)$:

$$\mathrm{FOU}(\tilde{B}(y|x_1')) = [\underline{\mu}_{\tilde{B}}(y|x_1'),\ \bar{\mu}_{\tilde{B}}(y|x_1')] \tag{9.48}$$

Comparing $\mathrm{FOU}(\tilde{B}(y|x_1'))$ in (9.48) and (9.31), observe that they are the same. Substituting (9.46) and (9.47) into (9.48), one obtains $(y \in Y)$:

$$\begin{aligned}\mathrm{FOU}(\tilde{B}(y|x')) &= [\underline{\mu}_{\tilde{F}_1}(x_1')\bigstar\underline{\mu}_{\tilde{G}}(y),\ \bar{\mu}_{\tilde{F}_1}(x_1')\bigstar\bar{\mu}_{\tilde{G}}(y)] \\ &\equiv [\underline{f}^1(x_1')\bigstar\underline{\mu}_{\tilde{G}}(y),\ \bar{f}^1(x_1')\bigstar\bar{\mu}_{\tilde{G}}(y)]\end{aligned} \tag{9.49}$$

where $\underline{f}^1(x_1') = \underline{\mu}_{\tilde{F}_1}(x_1')$ and $\bar{f}^1(x_1') = \bar{\mu}_{\tilde{F}_1}(x_1')$, in agreement with (9.25) and (9.26), respectively, when $p = 1$.

By this approach, the results that are in Corollaries 9.1 and 9.2 have been obtained using type-1 mathematics.

The extensions of this example to multiple antecedents and multiple rules can be found in Mendel et al. (2006, 2014, Chap. 3).

When Mendel et al. (2006) was published, it was felt that deriving all of the IT2 fuzzy system formulas using type-1 mathematics was a very good thing because it would let someone already familiar with T1 fuzzy systems learn about IT2 fuzzy systems very easily (some might say "painlessly"). At that time, everyone was focusing on IT2 fuzzy systems and the horizontal slice representation of a GT2 FS did not exist. Much has happened since 2006, and with the renewed study of GT2 fuzzy systems, this author no longer feels that this type-1 mathematical approach to deriving the IT2 fuzzy system formulas is as valuable as it once was. Expanding fuzzy sets using embedded fuzzy sets is arguably very tedious and the resulting derivations take about the same space as do the derivations that use the extended sup-star composition. Once one agrees to use the extended sup-star composition, then the resulting derivations are arguably very "clean."

9.4.2.2 Type-1 Non-singleton Fuzzifier[7]

For type-1 non-singleton fuzzification (Definition 9.5) the join and meet operations in (9.23) are not as easy to evaluate as in the singleton fuzzification case, because $\mu_{X_i}(x_i|x_1')$ is nonzero in a region about x'. The result is another corollary to Theorem 9.1:

[7]Readers who are not interested in non-singleton fuzzification can immediately go to Sect. 9.4.2.4.

Corollary 9.3 *For* type-1 non-singleton fuzzification (Liang and Mendel 2000a):

$$\begin{cases} \text{IT2 Mamdani fuzzy system:} & \mu_{\tilde{B}^l(y|\mathbf{x}')} = \left(1/[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]\right) \sqcap \mu_{\tilde{G}^l(y)} \quad y \in Y \\ \text{IT2 TSK fuzzy system:} & \mu_{\tilde{B}^l(\mathbf{x}')} = 1/[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')] \quad \text{when } y = g^l(\mathbf{x}') \end{cases} \tag{9.50}$$

where

$$\underline{f}^l(\mathbf{x}') = T_{i=1}^p \underline{f}_i^l = T_{i=1}^p \max_{x_i \in X_i} \mu_{X_i}(x_i|x_i') \star \underline{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.51}$$

$$\bar{f}^l(\mathbf{x}') = T_{i=1}^p \bar{f}_i^l = T_{i=1}^p \max_{x_i \in X_i} \mu_{X_i}(x_i|x_i') \star \bar{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.52}$$

The reader should compare Corollaries 9.1 and 9.3 to see that (9.50) is exactly the same as (9.24), and it is only the two endpoints of the firing intervals that are different. Now both $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ require computing a max-star composition between T1 FSs, similar to (3.21) in a type-1 fuzzy system.

Proof The starting point is (9.18), expressed as in (9.20) and (9.19), where the latter two equations are repeated here for the convenience of the readers:

$$F^l(\mathbf{x}') = \sqcap_{i=1}^p F^l(x_i') \tag{9.53}$$

$$F^l(x_i') \equiv \sqcup_{x_i \in X_i} \left(\mu_{\tilde{X}_i(x_i|x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)}\right) \tag{9.54}$$

Let

$$\mu_{\tilde{Q}_i^l}(x_i|x_i') \equiv \mu_{\tilde{X}_i(x_i\ |x_i')} \sqcap \mu_{\tilde{F}_i^l(x_i)} \tag{9.55}$$

so that

$$F^l(x_i') \equiv \sqcup_{x_i \in X_i} \mu_{\tilde{Q}_i^l}(x_i|x_i') \tag{9.56}$$

To compute $\mu_{\tilde{Q}_i^l}(x_i|x_i')$ for type-1 non-singleton fuzzification, $\mu_{\tilde{X}_i(x_i|x_i')}$ is expressed as the following IT2 FS in which the lower and upper MFs are the same:

$$\mu_{\tilde{X}_i(x_i|x_i')} \equiv 1/[\mu_{X_i}(x_i|x_i'),\ \mu_{X_i}(x_i|x_i')] \tag{9.57}$$

Additionally,

$$\mu_{\tilde{F}_i^l(x_i)} = 1/[\underline{\mu}_{\tilde{F}_i^l}(x_i),\ \bar{\mu}_{\tilde{F}_i^l}(x_i)] \tag{9.58}$$

Applying Theorem 7.12 to (9.55), one finds:

$$\mu_{\tilde{Q}_i^l}(x_i|x_i') = 1/[\mu_{X_i}(x_i|x_i') \bigstar \underline{\mu}_{\tilde{F}_i^l}(x_i),\ \mu_{X_i}(x_i|x_i') \bigstar \bar{\mu}_{\tilde{F}_i^l}(x_i)] \tag{9.59}$$

which can be re-expressed as

$$\mu_{\tilde{Q}_i^l}(x_i|x_i') = 1/[\underline{\mu}_{\tilde{Q}_i^l}(x_i|x_i'),\ \bar{\mu}_{\tilde{Q}_i^l}(x_i|x_i')] \tag{9.60}$$

where

$$\underline{\mu}_{\tilde{Q}_i^l}(x_i|x_i') = \mu_{X_i}(x_i|x_i') \bigstar \underline{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.61}$$

$$\bar{\mu}_{\tilde{Q}_i^l}(x_i|x_i') = \mu_{X_i}(x_i|x_i') \bigstar \bar{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.62}$$

In (9.56), at each $x_i \in X_i$, $\mu_{\tilde{Q}_i^l}(x_i|x_i')$ is a type-1 interval fuzzy number. Applying Theorem 7.11 to (9.56), using (9.60), one finds

$$F^l(x_i') = 1/[\underline{f}_i^l(x_i'),\ \bar{f}_i^l(x_i')] \tag{9.63}$$

where

$$\underline{f}_i^l(x_i') = \max_{x_i \in X_i} \underline{\mu}_{\tilde{Q}_i^l}(x_i|x_i') = \max_{x_i \in X_i} \mu_{X_i}(x_i|x_i') \bigstar \underline{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.64}$$

$$\bar{f}_i^l(x_i') = \max_{x_i \in X_i} \bar{\mu}_{\tilde{Q}_i^l}(x_i|x_i') = \max_{x_i \in X_i} \mu_{X_i}(x_i|x_i') \bigstar \bar{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.65}$$

Finally, applying Theorem 7.12 to (9.53), using (9.63), one finds

$$F^l(\mathbf{x}') = 1/[\underline{f}^l(\mathbf{x}'),\ \bar{f}^l(\mathbf{x}')] \tag{9.66}$$

where $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ are given in (9.51) and (9.52), respectively. This completes the proof of Corollary 9.3.

Let $\underline{x}_{i,\max}^l$ and $\bar{x}_{i,\max}^l$ denote the values of x_i that are associated with $\max_{x_i \in X_i} \underline{\mu}_{\tilde{Q}_i^l}(x_i|x_i')$ and $\max_{x_i \in X_i} \bar{\mu}_{\tilde{Q}_i^l}(x_i|x_i')$, respectively, so that

$$\underline{f}_i^l(x_i') = \underline{\mu}_{\tilde{Q}_i^l}(\underline{x}_{i,\max}^l|x_i') \tag{9.67}$$

$$\bar{f}_i^l(x_i') = \bar{\mu}_{\tilde{Q}_i^l}(\bar{x}_{i,\max}^l|x_i') \tag{9.68}$$

This means, of course, that $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ in (9.51) and (9.52) can be re-expressed as

$$\underline{f}^l(\mathbf{x}') = T_{i=1}^p \underline{f}_i^l(x_i') = T_{i=1}^p \underline{\mu}_{\tilde{Q}_i^l}(\underline{x}_{i,\max}^l | x_i') \tag{9.69}$$

$$\bar{f}^l(\mathbf{x}') = T_{i=1}^p \bar{f}_i^l(x_i') = T_{i=1}^p \bar{\mu}_{\tilde{Q}_i^l}(\bar{x}_{i,\max}^l | x_i') \tag{9.70}$$

To see the forest from the trees, so-to-speak, all of this is summarized next as a *four-step procedure to compute the firing interval* $[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]$:

1. Choose a t-norm (product or minimum) and create the functions $(i = 1, \ldots, p)$ $\underline{\mu}_{\tilde{Q}_i^l}(x_i|x_i')$ and $\bar{\mu}_{\tilde{Q}_i^l}(x_i|x_i')$ using (9.61) and (9.62), respectively.
2. Compute $(i = 1, \ldots, p)$ $\underline{x}_{i,\max}^l$ and $\bar{x}_{i,\max}^l$ by maximizing $\underline{\mu}_{\tilde{Q}_i^l}(x_i|x_i')$ and $\bar{\mu}_{\tilde{Q}_i^l}(x_i|x_i')$, respectively.
3. Evaluate $(i = 1, \ldots, p)$ $\underline{f}_i^l(x_i')$ and $\bar{f}_i^l(x_i')$ using (9.67) and (9.68), respectively.
4. Compute $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ using (9.69) and (9.70), respectively.

Corollary 9.4 *For* type-1 non-singleton fuzzification *and an* IT2 Mamdani fuzzy system (Liang and Mendel 2000a) *the MF for a* fired-rule output set, $\tilde{B}^l$, *is* $(l = 1, \ldots, M$ and $y \in Y)$:

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = 1/[\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}'),\ \bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}')] = 1/\mathrm{FOU}(\tilde{B}^l(y|\mathbf{x}')) \tag{9.71}$$

where

$$\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}') = \underline{f}^l(\mathbf{x}') \bigstar \underline{\mu}_{\tilde{G}^l}(y) \tag{9.72}$$

$$\bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}') = \bar{f}^l(\mathbf{x}') \bigstar \bar{\mu}_{\tilde{G}^l}(y) \tag{9.73}$$

in which $\underline{f}^l(\mathbf{x}')$ *and* $\bar{f}^l(\mathbf{x}')$ *are given in (9.51) and (9.52), respectively.*

Comparing Corollaries 9.4 and 9.2, observe that it is only the formulas used to compute $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ that are different.

It was mentioned below (3.24) that a non-singleton (type-1) fuzzy system first *pre-filters* its input $\mathbf{x}$, transforming it to $\mathbf{x}_{\max}^l$. A type-1 non-singleton IT2 fuzzy system also pre-filters its input $\mathbf{x}$, but it uses *two pre-filters*, simultaneously transforming it to $\underline{\mathbf{x}}_{\max}^l$ and $\bar{\mathbf{x}}_{\max}^l$.

Example 9.5 Here pictorial descriptions of the above four-step procedure are obtained for an IT2 Mamdani fuzzy system that uses either the minimum or product t-norms. As such, it is similar to Example 9.3, and will let pictorial descriptions for a type-1 non-singleton IT2 fuzzy system be contrasted with those for a singleton IT2 fuzzy system. It will also help to better understand the flow of rule-uncertainties

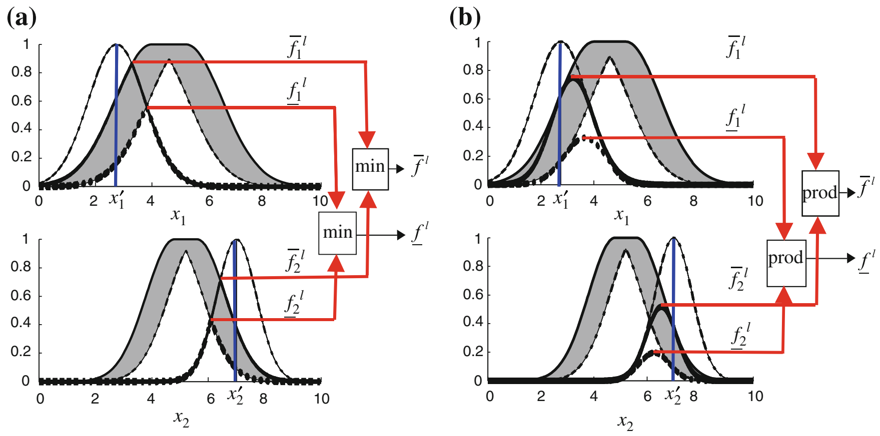


Fig. 9.5 Pictorial description of input, antecedent operations, and firing interval operations for an IT2 fuzzy system. Type-1 non-singleton fuzzification with: **a** minimum t-norm and **b** product t-norm (Liang and Mendel 2000a, p. 540; © 2000 IEEE)

combined with type-1 input uncertainties through a type-1 non-singleton IT2 fuzzy system.

Figure 9.5 depicts input, antecedent, and firing interval computations for a two-antecedent–single consequent rule, type-1 non-singleton fuzzification, and minimum or product t-norms. In both cases, the firing strength is again an interval $[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]$, where $\underline{f}^l(\mathbf{x}') = \underline{f}_1^l(\mathbf{x}') \bigstar \underline{f}_2^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}') = \bar{f}_1^l(\mathbf{x}') \bigstar \bar{f}_2^l(\mathbf{x}')$, in which $\underline{f}_i^l$ is the maximum of the firing strength between the t-norm of $\mu_{X_i}(x_i|x_i')$ and the LMF $\underline{\mu}_{\tilde{F}_i^l}(x_i)$, and $\bar{f}_i^l$ is the maximum of the firing strength between the t-norm of $\mu_{X_i}(x_i|x_i')$ and the UMF $\bar{\mu}_{\tilde{F}_i^l}(x_i)$ ($i = 1, 2$). Observe that $\mu_{X_i}(x_i|x_i')$ is centered at $x_i = x_i'$. These t-norms are shown as heavy curves in Fig. 9.5a, b. From these heavy curves it is easy to pick off their maxima. As in singleton fuzzification, regardless of the t-norm, the result of input and antecedent operations is an interval—the *firing interval*.

The results depicted in Fig. 9.3, $\mu_{\tilde{B}^l(y|\mathbf{x}')}$ $(y \in Y)$ for a two-rule singleton IT2 fuzzy system, remain pictorially the same for the type-1 non-singleton IT2 fuzzy system (the numerical values for $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ are different); however, type-1 non-singleton fuzzification leads to a firing interval that includes the *additional uncertainty* of the type-1 non-singleton inputs.

Example 9.6 Here the results that were presented in Example 9.5 are quantified, but for the case when the antecedent MFs are Gaussian primary MFs with uncertain standard deviations (as in Fig. 6.16) and the inputs are modeled as type-1 Gaussian fuzzy numbers, as in Fig. 9.6. $\bar{f}_i^l$ and $\underline{f}_i^l$ will be computed for this case.

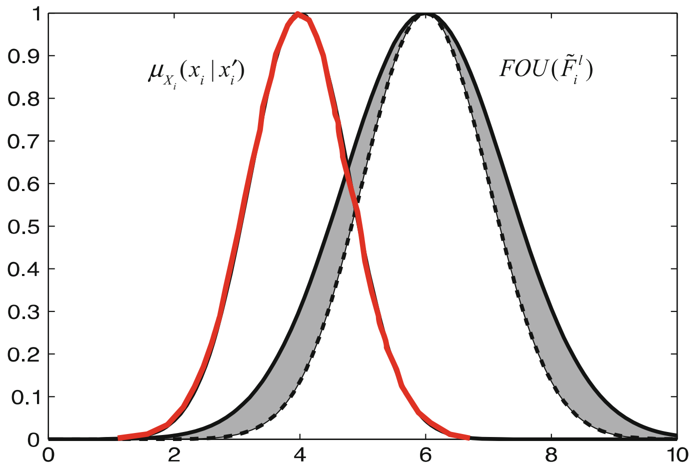


Fig. 9.6 The MFs for Example 9.6. For FOU($\tilde{F}_i^l$), the *thick solid line* denotes its UMF and the *thick dashed line* denotes its LMF. The *red* MF for the input is centered at x_i'

The MF for X_i is

$$\mu_{X_i}(x_i|x_i') = \exp\left[-\frac{1}{2}\left((x_i - x_i')/\sigma_{X_i}\right)^2\right] \tag{9.74}$$

The lower and upper antecedent MFs are $\underline{\mu}_{\tilde{F}_k^l}(x_k)$ and $\bar{\mu}_{\tilde{F}_k^l}(x_k)$, respectively where [see (6.24) and (6.23)]:

$$\underline{\mu}_{\tilde{F}_i^l}(x_i) = \exp\left[-\frac{1}{2}\left((x_i - m_i^l)/\sigma_{i1}^l\right)^2\right] \tag{9.75}$$

$$\bar{\mu}_{\tilde{F}_i^l}(x_i) = \exp\left[-\frac{1}{2}\left((x_i - m_i^l)/\sigma_{i2}^l\right)^2\right] \tag{9.76}$$

$\underline{f}_i^l(x_i')$ and $\bar{f}_i^l(x_i')$ both require the calculation of the maximum of a product or minimum of two Gaussian MFs. Such calculations were carried out in Example 3.7 for the product t-norm and in Exercise 2.40 for the minimum t-norm (see, also, footnote 11 in Chap. 3). The rest of this example only carries out the computations for the product t-norm. Exercise 9.4 asks the reader to carry them out for the minimum t-norm. Using (3.28), it follows that:

$$\underline{x}_{i,\max}^l = (\sigma_{X_i}^2 m_i^l + \sigma_{i1}^{l2} x_i')/(\sigma_{X_i}^2 + \sigma_{i1}^{l2}) \tag{9.77}$$

$$\bar{x}_{i,\max}^l = (\sigma_{X_i}^2 m_i^l + \sigma_{i2}^{l2} x_i')/(\sigma_{X_i}^2 + \sigma_{i2}^{l2}) \tag{9.78}$$

Using (3.29) in (9.67) and (9.68), along with (9.61) and (9.62), it then follows that:

$$\underline{f}_i^l = \underline{\mu}_{\tilde{Q}_i^l}(\underline{x}_{i,\max}^l|x_i') = \mu_{X_i}(\underline{x}_{i,\max}^l|x_i')\,\underline{\mu}_{\tilde{F}_i^l}(\underline{x}_{i,\max}^l) = \exp\left[-\frac{1}{2}\left((m_i^l - x_i')^2/(\sigma_{X_i}^2 + \sigma_{i1}^{l2})\right)\right] \tag{9.79}$$

$$\bar{f}_i^l = \bar{\mu}_{\tilde{Q}_i^l}(\bar{x}_{i,\max}^l|x_i') = \mu_{X_i}(\bar{x}_{i,\max}^l|x_i')\,\bar{\mu}_{\tilde{F}_i^l}(\bar{x}_{i,\max}^l) = \exp\left[-\frac{1}{2}\left((m_i^l - x_i')^2/(\sigma_{X_i}^2 + \sigma_{i2}^{l2})\right)\right] \tag{9.80}$$

Example 3.7 proved that the firing level is larger in the non-singleton case than it is in the singleton case. In a similar manner, one can prove that *the firing interval is larger in the type-1 non-singleton case than it is in the type-2 singleton case* (Exercise 9.5) (see, also, Example 9.9).

Mendel (2001, Chap. 11) has another example that is similar to the present example in which the antecedent MFs are Gaussian primary MFs with uncertain means (see Example 6.17), and the inputs are again modeled as type-1 Gaussian fuzzy numbers. The computations of $\underline{f}_i^l(x_i')$ and $\bar{f}_i^l(x_i')$ in that example are much more complicated than the ones in the present example, because the formulas for the lower and upper MFs change with each x_i, something that does not occur when the antecedent MFs are Gaussian primary MFs with uncertain standard deviations. They are not included in this book because, in the opinion of this author, they are too complicated to be used in real-world applications.

9.4.2.3 IT2 Non-singleton Fuzzifier

For IT2 non-singleton fuzzification (Definition 9.6), the join and meet operations in (9.23) are more difficult to evaluate than in the type-1 non-singleton fuzzification case, because the input MF is now an IT2 FS, usually an IT2 fuzzy number (Definition 6.22). The result is another corollary to Theorem 9.1:

Corollary 9.5 *For* IT2 non-singleton fuzzification (Liang and Mendel 2000a):

$$\begin{cases} \text{IT2 Mamdani fuzzy system:} & \mu_{\tilde{B}^l(y|\mathbf{x}')} = \left(1/[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]\right) \sqcap \mu_{\tilde{G}^l(y)} \quad y \in Y \\ \text{IT2 TSK fuzzy system:} & \mu_{\tilde{B}^l(\mathbf{x}')} = 1/[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')] \text{ when } y = g^l(\mathbf{x}') \end{cases} \tag{9.81}$$

where

$$\underline{f}^l(\mathbf{x}') = T_{i=1}^p \underline{f}_i^l = T_{i=1}^p \max_{x_i \in X_i} \underline{\mu}_{\tilde{X}_i}(x_i|x_i') \bigstar \underline{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.82}$$

$$\bar{f}^l(\mathbf{x}') = T_{i=1}^p \bar{f}_i^l = T_{i=1}^p \max_{x_i \in X_i} \bar{\mu}_{\tilde{X}_i}(x_i|x_i') \bigstar \bar{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.83}$$

The reader should compare Corollaries 9.5 and 9.3 to see that (9.81) is exactly the same as (9.50), and again it is only the two endpoints of the firing intervals that are different. Again, both $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ require computing a max-star composition between T1 FSs, similar to (3.21) in a type-1 fuzzy system, but $\underline{f}^l(\mathbf{x}')$ involves $\underline{\mu}_{\tilde{X}_i}(x_i|x_i')$ whereas $\bar{f}^l(\mathbf{x}')$ involves $\bar{\mu}_{\tilde{X}_i}(x_i|x_i')$, whereas in Corollary 9.3 they both involved the same type-1 MF $\mu_{X_i}(x_i|x_i')$.

Proof Because the proof of this corollary is very similar to the proof of Corollary 9.3, it is left to the reader in Exercise 9.6. In the proof, the following will be defined:

$$\underline{\mu}_{\tilde{Q}_i^l}(x_i|x_i') = \underline{\mu}_{\tilde{X}_i}(x_i|x_i')\bigstar\underline{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.84}$$

$$\bar{\mu}_{\tilde{Q}_i^l}(x_i|x_i') = \bar{\mu}_{\tilde{X}_i}(x_i|x_i')\bigstar\bar{\mu}_{\tilde{F}_i^l}(x_i) \tag{9.85}$$

It is these two functions that have to be maximized in order to compute $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ in (9.82) and (9.83), respectively.

The four-step procedure that is stated below (9.70) applies here as well, modulo obvious changes in equation numbers.

Corollary 9.6 *For* IT2 non-singleton fuzzification *and an* IT2 Mamdani fuzzy system (Liang and Mendel 2000a) *the MF for a* fired-rule output set, $\tilde{B}^l$, *is* $(l = 1,\ldots,M$ and $y \in Y)$:

$$\mu_{\tilde{B}^l(y|\mathbf{x}')} = 1/[\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}'),\ \bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}')] = 1/\mathrm{FOU}(\tilde{B}^l(y|\mathbf{x}')) \tag{9.86}$$

where

$$\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}') = \underline{f}^l(\mathbf{x}')\bigstar\underline{\mu}_{\tilde{G}^l}(y) \tag{9.87}$$

$$\bar{\mu}_{\tilde{B}^l}(y\ |\mathbf{x}') = \bar{f}^l(\mathbf{x}')\bigstar\bar{\mu}_{\tilde{G}^l}(y) \tag{9.88}$$

in which $\underline{f}^l(\mathbf{x}')$ *and* $\bar{f}^l(\mathbf{x}')$ *are given in (9.82) and (9.83), respectively.*

Comparing Corollaries 9.6 and 9.4, observe that it is only the formulas used to compute $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ that are different.

Example 9.7 This example is similar to Example 9.5, and will let pictorial descriptions for an IT2 non-singleton fuzzy system be contrasted with those for a type-1 non-singleton IT2 fuzzy system. It will also help to better understand the flow of rule-uncertainties combined with IT2 input uncertainties through an IT2 non-singleton fuzzy system.

Figure 9.7 depicts input, antecedent, and firing interval computations for a two-antecedent–single consequent rule, IT2 non-singleton fuzzification, and minimum or product t-norms. In both cases, the firing strength is again an interval

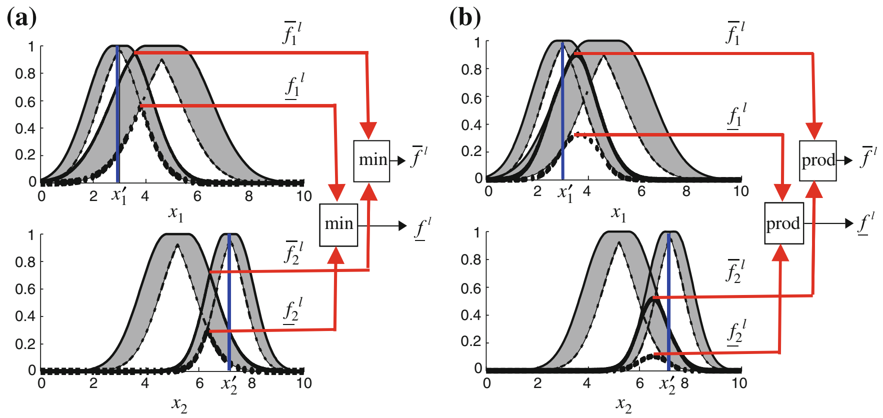


Fig. 9.7 Pictorial description of input, antecedent operations, and firing interval operations for an IT2 fuzzy system. IT2 non-singleton fuzzification with: **a** minimum t-norm and **b** product t-norm (Liang and Mendel 2000a, p. 540; © 2000 IEEE)

$[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')]$, where $\underline{f}^l(\mathbf{x}') = \underline{f}_1^l(\mathbf{x}')\star\underline{f}_2^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}') = \bar{f}_1^l(\mathbf{x}')\star\bar{f}_2^l(\mathbf{x}')$, in which $\underline{f}_i^l(\mathbf{x}')$ is the maximum of the firing strength between the t-norm of the LMFs $\underline{\mu}_{\tilde{X}_i}(x_i)$ and $\underline{\mu}_{\tilde{F}_i^l}(x_i)$, and $\bar{f}_i^l(\mathbf{x}')$ is the maximum of the firing strength between the t-norm of the UMFs $\bar{\mu}_{\tilde{X}_i}(x_i)$ and $\bar{\mu}_{\tilde{F}_i^l}(x_i)$ (i = 1, 2). Observe that $\mu_{\tilde{X}_i}(x_i)$ is again centered at $x_i = x_i'$. These t-norms are shown as heavy curves in Fig. 9.7a, b. From these heavy curves, it is again easy to pick off their maximum values. As in singleton and type-1 non-singleton fuzzification, regardless of the t-norm, the result of input and antecedent operations is an interval—the *firing interval*.

The results depicted in Fig. 9.3, $\mu_{\tilde{B}^l(y|\mathbf{x}')}$ $(y \in Y)$ for a two-rule singleton IT2 fuzzy system, remain pictorially the same for the IT2 non-singleton fuzzy system (the numerical values for $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$ are different); however, IT2 non-singleton fuzzification leads to a firing interval that includes the *additional uncertainty* of the IT2 non-singleton inputs.

Example 9.8 In Chap. 3 pictures like the ones in Figs. 9.2, 9.5 and 9.7 were given for singleton type-1 (Fig. 3.5) and non-singleton type-1 (Fig. 3.8) fuzzy systems. It is instructive to compare all of these results; hence, these results are collected for product t-norm in Fig. 9.8. The comparisons for minimum t-norm are similar. Observe that *uncertainties increase either the firing level* (compare Fig. 9.10a, b) *or the firing interval* (compare Fig. 9.8c–e). For the type-2 cases, *as more uncertainties are included* (i.e., in going from singleton IT2 fuzzy system to a type-1 non-singleton IT2 fuzzy system to an IT2 non-singleton IT2 fuzzy system), *the larger the firing interval becomes*. Example 9.9 explores some of this mathematically.

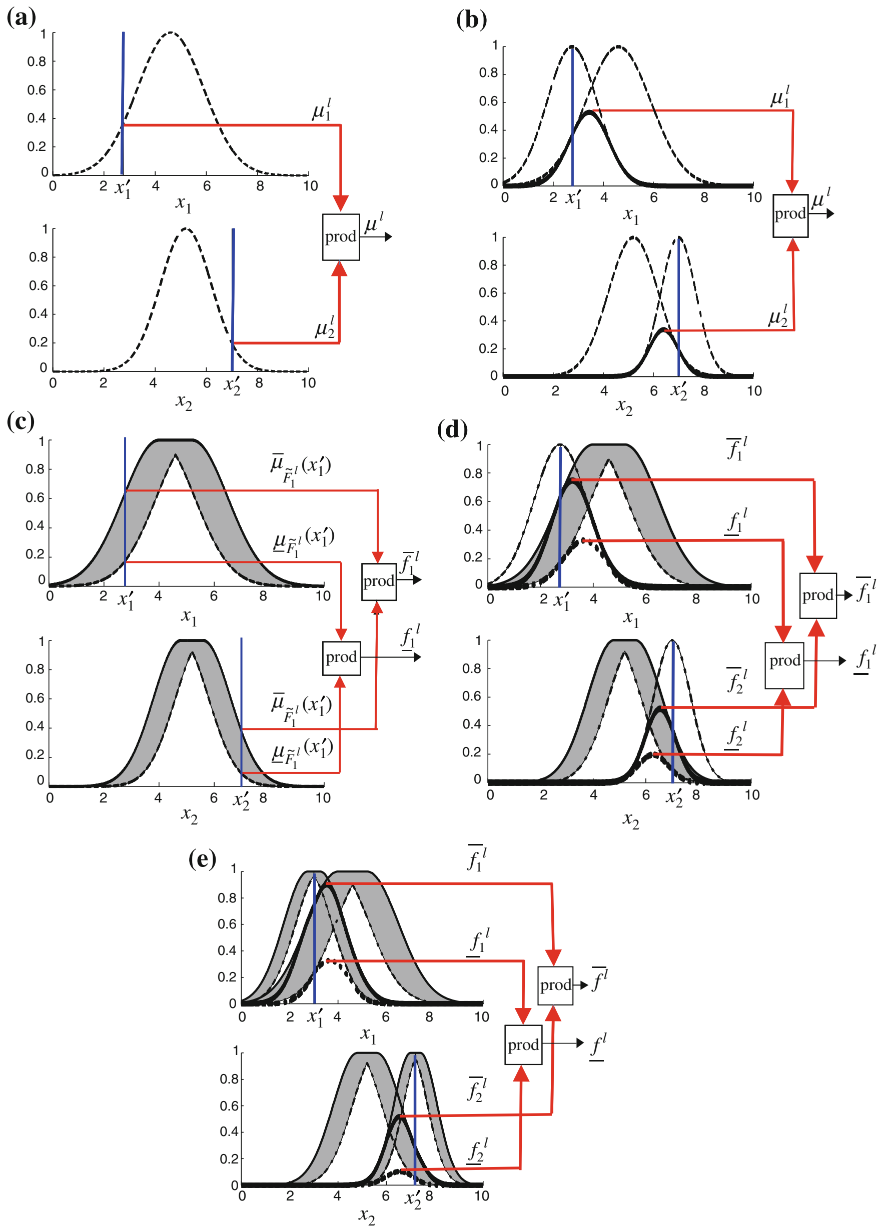


Fig. 9.8 Pictorial description of input, antecedent operations, and firing quantity using product t-norm for fuzzy systems: **a** singleton T1, **b** non-singleton T1, **c** singleton IT2, **d** T1 non-singleton IT2, and **e** IT2 non-singleton IT2

Example 9.9 This example is a generalization of Example 9.6. The situation now is the one that is depicted in Fig. 9.9. Here

$$\underline{\mu}_{\tilde{X}_i}(x_i|x_i') = \exp\left[-\frac{1}{2}\left((x_i - x_i')/\sigma_{i1}\right)^2\right] \tag{9.89}$$

$$\bar{\mu}_{\tilde{X}_i}(x_i|x_i') = \exp\left[-\frac{1}{2}\left((x_i - x_i')/\sigma_{i2}\right)^2\right] \tag{9.90}$$

and $\underline{\mu}_{\tilde{F}_k^l}(x_k)$ and $\bar{\mu}_{\tilde{F}_k^l}(x_k)$ are given in (9.75) and (9.76), respectively. Using (3.28), it follows that

$$\underline{x}_{i,\max}^l = (\sigma_{i1}^2 m_i^l + \sigma_{i1}^{l2} x_i')/(\sigma_{i1}^2 + \sigma_{i1}^{l2}) \tag{9.91}$$

$$\bar{x}_{i,\max}^l = (\sigma_{i2}^2 m_i^l + \sigma_{i2}^{l2} x_i')/(\sigma_{i2}^2 + \sigma_{i2}^{l2}) \tag{9.92}$$

Using (3.29) in (9.82) and (9.83), it then follows that:

$$\underline{f}_i^l = \underline{\mu}_{\tilde{Q}_i^l}(\underline{x}_{i,\max}^l|x_i') = \mu_{\tilde{X}_i}(\underline{x}_{i,\max}^l|x_i')\underline{\mu}_{\tilde{F}_i^l}(\underline{x}_{i,\max}^l) = \exp\left[-\frac{1}{2}\left((m_i^l - x_i')^2/(\sigma_{i1}^2 + \sigma_{i1}^{l2})\right)\right] \tag{9.93}$$

$$\bar{f}_i^l = \bar{\mu}_{\tilde{Q}_i^l}(\bar{x}_{i,\max}^l|x_i') = \mu_{\tilde{X}_i}(\bar{x}_{i,\max}^l|x_i')\,\bar{\mu}_{\tilde{F}_i^l}(\bar{x}_{i,\max}^l) = \exp\left[-\frac{1}{2}\left((m_i^l - x_i')^2/(\sigma_{i2}^2 + \sigma_{i2}^{l2})\right)\right] \tag{9.94}$$

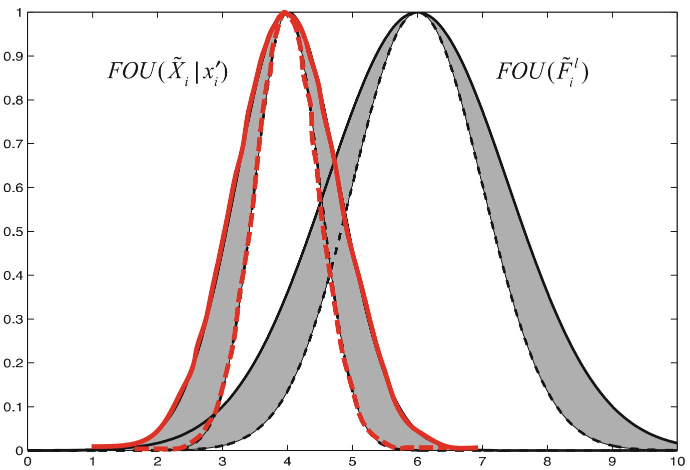


Fig. 9.9 The MFs for Example 9.9. For both FOUs, the *thick solid line* denotes the UMF and the *thick dashed line* denotes the LMF. The *red* MFs for the input are centered at x_i'

These results should be compared with those in (9.79) and (9.80), respectively. Because $\sigma_{i1}^2 < \sigma_{X_i}^2 \le \sigma_{i2}^2$, $\sigma_{i1}^2 < \sigma_{X_i}^2$ and $\sigma_{i2}^2 > \sigma_{X_i}^2$, so that $\underline{f}_i^l\big|_{(9.93)} < \underline{f}_i^l\big|_{(9.79)}$ and $\bar{f}_i^l\big|_{(9.94)} > \bar{f}_i^l\big|_{(9.80)}$, which proves (at least for this example) the statement made at the end of Example 9.8, that as more uncertainties are included, the larger the firing interval becomes.

Mendel (2001, Chap. 12) has another example that is similar to the present example in which the antecedent MFs are Gaussian primary MFs with uncertain means (see Example 6.17), and the inputs are modeled as IT2 Gaussian fuzzy numbers with uncertain standard deviations (see Example 6.16). The computations of $\underline{f}_i^l(x_i')$ and $\bar{f}_i^l(x_i')$ in that example are much more complicated than the ones in the present example, because the formulas for the lower and upper MFs change with each x_i, in a very complicated way. They are not included in this book because, in the opinion of this author, they are much too complicated to be used in real-world applications.

9.4.2.4 IT2 Rule Partitions

The firing intervals partition $X_1 \times X_2 \times \cdots \times X_p$ in two interesting ways, and since the formula for the firing interval is exactly the same for IT2 Mamdani and TSK fuzzy systems these partitions are the same for both kinds of fuzzy systems.

Definition 9.8 An *IT2 first-order rule partition* of $X_1 \times X_2 \times \cdots \times X_p$ is a collection of nonoverlapping rectangles [information granules (see Definition 6.16)] in each of which a fixed number of rules is fired, where that number is found by examining the antecedent FOUs simultaneously.

Although the wording of this definition is very similar to the wording for a type-1 first-order rule partition of $X_1 \times X_2 \times \cdots \times X_p$, given in Definition 3.6, it is the way in which the IT2 first-order rule partitions are found that is different from the way in which a type-1 first-order rule partition of $X_1 \times X_2 \times \cdots \times X_p$ is found. This will be explained in Example 9.10.

Definition 9.9 An *IT2 second-order rule partition* of $X_1 \times X_2 \times \cdots \times X_p$ occurs when the LMF or the UMF of an FOU of an IT2 FS that is associated with either x_1, or x_2, or …, or x_p changes its mathematical formula within an IT2 first-order rule partition.

Another way to state this is that for an IT2 fuzzy system the use of its lower and upper MFs to compute the firing interval within an IT2 first-order rule partition may be *adaptive* to the locations within that partition (Wu 2012).

IT2 first-order and second-order rule partitions are consequences of second-order uncertainty partitions (Definition 1.3).

Example 9.10 This example is the IT2 version of Example 3.3. Suppose, as in that example, rules have two antecedents, x_1 and x_2, and that each of them is again described by the three linguistic terms, *low* $(\tilde{L})$, *moderate* $(\tilde{M})$ and *high* $(\tilde{H})$, so that there are $3 \times 3 = 9$ rules. Now *low* and *high* are described by trapezoidal shoulder FOUs and *moderate* is described by an interior trapezoidal FOU, as depicted on Fig. 9.10. Observe that, as in Fig. 3.3, X_1 and X_2 are each *partitioned* into five rectangles, three of which have no overlapping FOUs and two of which have overlapping FOUs. Consequently, $X_1 \times X_2$ has 25 IT2 first-order rule partitions. The numbers 1, 2 or 4 that appear in each of these regions again denote how many two-antecedent rules are associated with that rectangle. For examples of the rules see Example 3.3, but now all of the T1 FSs are now replaced by IT2 FSs.

Comparing Figs. 9.10 and 3.3, one's first impression is that they look different. Most noticeable is that the rectangles for 2 and 4 rules in Fig. 9.10 are significantly larger than the comparable rectangles in Fig. 3.3. A closer comparisons of these two figures reveals that they have exactly the same number of rule partitions and that the rule-numbers in the rectangles of Fig. 9.10 are exactly the same as those in Fig. 3.3. In fact, Fig. 9.10 IT2 first-order rule partitions are obtained just by using the UMFs of the FOUs, which are type-1 fuzzy sets. So what's different?

Overlaying Figs. 3.3 and 9.10, one observes that in each of the rectangles of Fig. 9.10 where 2 rules are fired, 1 or 2 rules will be fired in Fig. 3.3 rectangle, and in each of the rectangles of Fig. 9.10 where 4 rules are fired, 1, 2, or 4 rules will be fired in Fig. 3.3 rectangle. This means that (except for rectangles where only 1 rule is fired) the IT2 fuzzy system is firing more rules over much more of $X_1 \times X_2$ than is a type-1 fuzzy system.

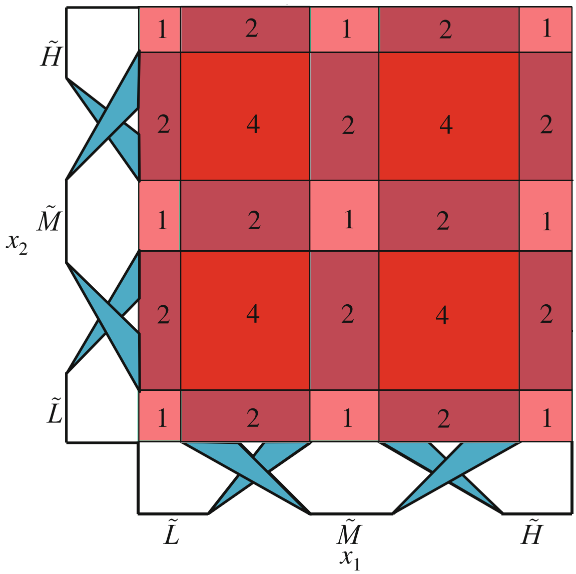


Fig. 9.10 IT2 first-order partitions of $X_1 \times X_2$ for two-antecedent rules when each antecedent is described by the three linguistic terms, *low* $(\tilde{L})$, *moderate* $(\tilde{M})$ and *high* $(\tilde{H})$. The number in each *shaded rectangle* gives the number of rules that are fired in it

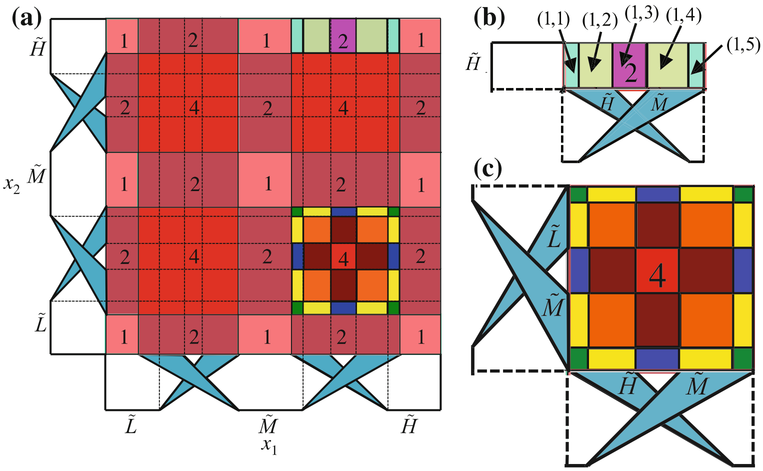


Fig. 9.11 **a** Construction of the IT2 second-order rule partition diagram of $X_1 \times X_2$ for Example 9.10 (the *dashed horizontal* and *vertical lines* occur where the LMF or the UMF of an FOU changes its mathematical formula); **b** blowup of the Fig. 9.11a *top* row and fourth rectangle from the *left*; and **c** blowup of the Fig. 9.11a fourth row *down* from the *top* and fourth rectangle from the *left*. In (**b**) and (**c**) the FOUs for x_1 and x_2 have been moved so that they are adjacent to the respective IT2 first-order rule partitions

Figure 9.11a shows the construction of an *IT2 second-order rule partition diagram*. Its dashed horizontal and vertical lines occur where either a LMF or an UMF of a FOU changes its mathematical formula. It is these additional lines that further partition the IT2 first-order rule partitions into IT2 second-order rule partitions. Note that *in this example* IT2 second-order rule partitions only occur in the IT2 first-order rule partition rectangles in which more than one rule is fired.

Consider the following two IT2 first-order partitions:

1. Observe in the top row and fourth rectangle from the left in Fig. 9.10, that two rules are activated, but different combinations of the lower and upper MFs of the IT2 FSs are used to compute a firing quantity (see Fig. 9.11a). In fact, there are five IT2 second-order rule partitions of the IT2 first-order rule partition (a blowup of this is given in Fig. 9.11b). In each of these IT2 second-order rule partitions the same lower and upper MFs for "x_2 is $\tilde{H}$" is used, but there are five different combinations of the overlapping portions of the lower and upper MFs for "x_1 is $\tilde{M}$" and "x_1 is $\tilde{H}$" that are used, namely:
   - $(\mathrm{LMF}(\tilde{M}), \mathrm{UMF}(\tilde{M});\ \mathrm{LMF}(\tilde{H}), \mathrm{UMF}(\tilde{H}))_{11} = (1, 1; 0, \mathrm{UMF}(\tilde{H}))$
   - $(\mathrm{LMF}(\tilde{M}), \mathrm{UMF}(\tilde{M});\ \mathrm{LMF}(\tilde{H}), \mathrm{UMF}(\tilde{H}))_{12} = (\mathrm{LMF}(\tilde{M}), \mathrm{UMF}(\tilde{M});$ $0, \mathrm{UMF}(\tilde{H}))$
   - $(\mathrm{LMF}(\tilde{M}), \mathrm{UMF}(\tilde{M});\ \mathrm{LMF}(\tilde{H}), \mathrm{UMF}(\tilde{H}))_{13} = (\mathrm{LMF}(\tilde{M}), \mathrm{UMF}(\tilde{M});$ $\mathrm{LMF}(\tilde{H}), \mathrm{UMF}(\tilde{H}))$

– $(\text{LMF}(\tilde{M}), \text{UMF}(\tilde{M});\ \text{LMF}(\tilde{H}), \text{UMF}(\tilde{H}))_{14} = (0, \text{UMF}(\tilde{M}); \text{LMF}(\tilde{H}), \text{UMF}(\tilde{H}))$
– $(\text{LMF}(\tilde{M}), \text{UMF}(\tilde{M});\ \text{LMF}(\tilde{H}), \text{UMF}(\tilde{H}))_{15} = (0, \text{UMF}(\tilde{M}); 1, 1)$

2. Observe in the next to the last row from the top and fourth rectangle from the left in Fig. 9.10, that four rules are activated but now different combinations of the lower and upper MFs of the IT2 FSs are used to compute a firing quantity (see Fig. 9.11a). In fact, there are 25 IT2 second-order rule partitions of the IT2 first-order rule partition (a blowup of this is given in Fig. 9.11c). In each of these IT2 second-order rule partitions there are different combinations of the overlapping portions of the lower and upper MFs for "x_2 is $\tilde{L}$" and "x_2 is $\tilde{M}$" and "x_1 is $\tilde{M}$" and "x_1 is $\tilde{H}$." The exact natures of these 25 IT2 second-order rule partitions are left to the reader (Exercise 9.7).

For the FOUs used in this example, whereas there are 25 IT2 first-order rule partitions, there are 135 IT2 second-order rule partitions (count them), and all of this has been accomplished with only 9 rules! The enumerations of these IT2 second-order rule partitions is unnecessary, i.e., the IT2 second-order rule partitions occur automatically simply by using overlapping FOUs.

Example 9.10 is unchanged for type-1 non-singleton fuzzification, or for that matter for any kind of fuzzification, because IT2 first- and second-order rule partitions only depend upon antecedent FOUs.

Note that Example 3.4, that counts the number of first-order rule partitions in $X_1 \times X_2$, is still valid for IT2 fuzzy systems, i.e., it gives the correct number of IT2 first-order partitions. Exercise 9.8 asks the reader to obtain formulas that give the number of IT2 first- and second-order rule partitions in $X_1 \times X_2 \times X_3$ for an IT2 fuzzy system.

Whereas it is the nature of the antecedent FOUs that establish the IT2 first- and second-order rule partitions, it is the nature of each rule's consequent that fills each rule partition. IT2 Zadeh rules fill each IT2 rule partition with a linguistic term (or its surrogate, e.g., the centroid of the FOU for that term), whereas an IT2 TSK rule fills each IT2 rule partition with a mathematical formula.

At the end of Sect. 3.4.2, it was mentioned that a type-1 fuzzy system is a variable structure system. Clearly, an IT2 fuzzy system is also a variable structure system, but with much greater variability than a type-1 fuzzy system.

MF uncertainty can be used to control *rule explosion*. In a type-1 fuzzy system, greater resolution is accomplished by partitioning the variables more finely, leading to more rules. Example 9.10 has already demonstrated that a very large number of IT2 second-order rule partitions can be created by using IT2 FSs, and it is this large number of IT2 second-order rule partitions that lets an IT2 fuzzy system achieve better resolution than a type-1 fuzzy system when both systems use the same number of rules.

9.5 Combining Fired Rule Output Sets on the Way to Defuzzification

Observe in Fig. 9.1 that the output(s) from the inference block go either to the type-reduction block or directly to the defuzzifier block. If more than one rule fires, one first needs to decide what to do with them (as was the situation for a type-1 fuzzy system) either on the way to type-reduction or to direct defuzzification. As pointed out in Sect. 3.5, there is no unique way to combine fired-rule output sets. In the rest of this section, two possibilities are described.[8]

9.5.1 Combining Using Set Theoretic Operations in an IT2 Mamdani Fuzzy System

Just as type-1 fired-rule output sets for a type-1 Mamdani fuzzy system can be combined by using the union operation, as explained in Sect. 3.5.1, IT2 fired-rule output sets for an IT2 Mamdani fuzzy system can also be combined by using the union operation; this involves computing the join of those sets.

Theorem 9.2 *Suppose that M_F of the M rules in an IT2 Mamdani fuzzy system fire, where $M_F \leq M$, and a combined output IT2 FS $\tilde{B}$ is obtained by combining the fired-rule output sets using the union operation; then $(y \in Y)$:*

$$\mu_{\tilde{B}(y|\mathbf{x}')} = 1\Big/[\underline{\mu}_{\tilde{B}}(y|\mathbf{x}'), \bar{\mu}_{\tilde{B}}(y|\mathbf{x}')] = 1/\text{FOU}(\tilde{B}(y|\mathbf{x}')) \tag{9.95}$$

where

$$\underline{\mu}_{\tilde{B}}(y|\mathbf{x}') = \vee_{l=1}^{M_F} \underline{f}^l(\mathbf{x}') \bigstar \underline{\mu}_{\tilde{G}^l}(y) \tag{9.96}$$

$$\bar{\mu}_{\tilde{B}}(y|\mathbf{x}') = \vee_{l=1}^{M_F} \bar{f}^l(\mathbf{x}') \bigstar \bar{\mu}_{\tilde{G}^l}(y) \tag{9.97}$$

In (9.96) and (9.97), to compute $\underline{f}^l(\mathbf{x}')$ and $\bar{f}^l(\mathbf{x}')$, use (9.25) and (9.26) for singleton fuzzification, (9.51) and (9.52) for T1 non-singleton fuzzification, and (9.82) and (9.83) for IT2 non-singleton fuzzification.

[8]For a type-1 fuzzy system there were three possible ways to combined fired rule output sets, one of which was combining by using a weighted combination. To-date, this approach has not been used for T2 fuzzy systems, due arguably to the additional complexity of having to add T2 FSs.

Proof The union of the M_F fired-rule IT2 output sets, $\tilde{B}^l$ $(l = 1, \ldots, M_F)$ is $(y \in Y)$:

$$\tilde{B} = \bigcup_{l=1}^{M_F} \tilde{B}^l \tag{9.98}$$

Applying Corollary 7.3 $M_F - 1$ times to (9.98), one obtains $(y \in Y)$:

$$\mu_{\tilde{B}} = 1 \,/\, [\vee_{l=1}^{M_F} \underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}'),\ \vee_{l=1}^{M_F} \bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}')] \equiv 1/[\underline{\mu}_{\tilde{B}}(y|\mathbf{x}'),\ \bar{\mu}_{\tilde{B}}(y|\mathbf{x}')] \tag{9.99}$$

The same looking formulas are given for $\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}')$ in Corollaries 9.2, 9.4 and 9.6, namely:

$$\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}') = \underline{f}^l(\mathbf{x}') \bigstar \underline{\mu}_{\tilde{G}^l}(y) \tag{9.100}$$

$$\bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}') = \bar{f}^l(\mathbf{x}') \bigstar \bar{\mu}_{\tilde{G}^l}(y) \tag{9.101}$$

Substituting (9.100) and (9.101) into (9.99), one obtains the results in (9.95)–(9.97).

Example 9.11 Figure 9.12 depicts the FOU for the combined output set, $\tilde{B}$, for a two-rule IT2 fuzzy system, where the fired-rule output sets are depicted in Fig. 9.3, and are combined as in (9.95)–(9.97) using the maximum for the disjunction. The results are fairly similar for both t-norms. The upper solid curve corresponds to $\max\left[\bar{f}^1(\mathbf{x}') \bigstar \bar{\mu}_{\tilde{G}^1}(y),\ \bar{f}^2(\mathbf{x}') \bigstar \bar{\mu}_{\tilde{G}^2}(y)\right]$ for $y \in Y$, whereas the lower dashed curve corresponds to $\max\left[\underline{f}^1(\mathbf{x}') \bigstar \underline{\mu}_{\tilde{G}^1}(y),\ \underline{f}^2(\mathbf{x}') \bigstar \underline{\mu}_{\tilde{G}^2}(y)\right]$ for $y \in Y$. FOU($\tilde{B}$) is the area between these two functions, which has been darkened in. Type-reduction would be applied to $\tilde{B}$ $y \in Y$.

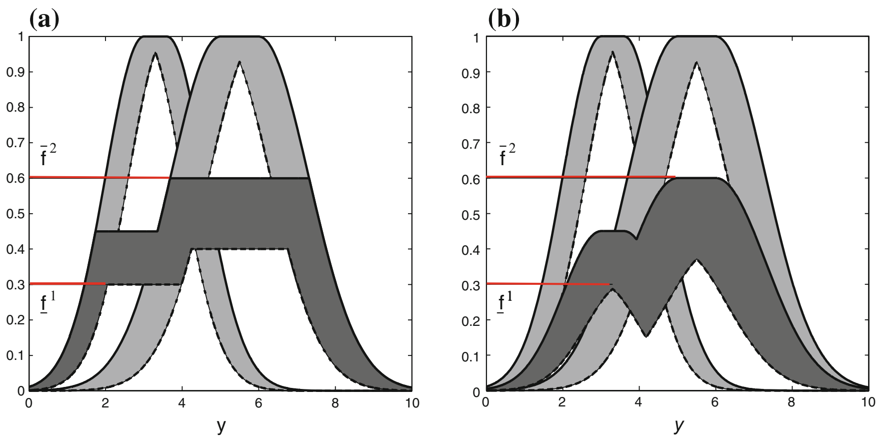


Fig. 9.12 Pictorial description of the combined two-rule fired-rule output sets depicted in: **a** Fig. 9.3a and **b** Fig. 9.3b

The reader should compare Fig. 9.12 with its respective type-1 counterpart in Fig. 3.9 to fully appreciate the way in which an IT2 fuzzy system has let the uncertainties about antecedent and consequent MFs flow from the input of the IT2 fuzzy system to its aggregated output FOU.

9.5.2 Combining During Defuzzification in an IT2 Mamdani Fuzzy System

In principle one can combine fired-rule output sets as just described; however, when an IT2 Mamdani fuzzy system is used in a real-world application, the issue of computational complexity becomes as important as it is when a type-1 fuzzy Mamdani system is used, especially if the application is one that requires real-time processing. For an IT2 Mamdani fuzzy system, taking the union of IT2 fired-rule output sets requires additional computation time and storage, which many practitioners have found to be very undesirable, and so an alternative to performing any kind of combining prior to defuzzification for such rules is to performing no combining. What this means is explained next in Sect. 9.6.

9.6 Type-Reduction + Defuzzification

Three kinds of type-reduction were described in Sects. 8.3.2–8.3.4 for an IT2 Mamdani fuzzy system. For self-containment in this chapter, portions of those sections are repeated here. In addition type-reduction is described for an IT2 TSK fuzzy system.

For any kind of IT2 fuzzy system, type-reduction leads to an interval type-1 fuzzy set, for which defuzzification is trivial, namely, it is taken as the average of the endpoints of the type-reduced set.

9.6.1 Centroid Type-Reduction + Defuzzification for an IT2 Mamdani Fuzzy System

Centroid type-reduction in an IT2 Mamdani fuzzy system is performed on $\mu_{\tilde{B}(y|\mathbf{x}')}$ in (9.95), to obtain $C_{\tilde{B}}$; it requires the entire FOU of $\tilde{B}$, or a sampled version of it, as its starting point, and is:

$$C_{\tilde{B}} = 1/[c_l(\tilde{B}),\ c_r(\tilde{B})] \tag{9.102}$$

In (9.102), $\tilde{B}$ is short for $\tilde{B}(y|\mathbf{x}')$, and $c_l(\tilde{B})$ and $c_l(\tilde{B})$ are computed as explained in Theorem 8.1, in which:

$$c_l(\tilde{B}) = \frac{\sum_{i=1}^{L} y_i \bar{\mu}_{\tilde{B}}(y_i|\mathbf{x}') + \sum_{i=L+1}^{N} y_i \underline{\mu}_{\tilde{B}}(y_i|\mathbf{x}')}{\sum_{i=1}^{L} \bar{\mu}_{\tilde{B}}(y_i|\mathbf{x}') + \sum_{i=L+1}^{N} \underline{\mu}_{\tilde{B}}(y_i|\mathbf{x}')} \tag{9.103}$$

$$c_r(\tilde{B}) = \frac{\sum_{i=1}^{R} y_i \underline{\mu}_{\tilde{B}}(y_i|\mathbf{x}') + \sum_{i=R+1}^{N} y_i \bar{\mu}_{\tilde{B}}(y_i|\mathbf{x}')}{\sum_{i=1}^{R} \underline{\mu}_{\tilde{B}}(y_i|\mathbf{x}') + \sum_{i=R+1}^{N} \bar{\mu}_{\tilde{B}}(y_i|\mathbf{x}')} \tag{9.104}$$

In (9.103) and (9.104), $\underline{\mu}_{\tilde{B}}(y_i|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}}(y_i|\mathbf{x}')$ are sampled values of $\underline{\mu}_{\tilde{B}}(y|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}}(y|\mathbf{x}')$, that are given in (9.96) and (9.97), respectively. EIASC or EKM algorithms can be used to compute $c_l(\tilde{B})$ and $c_l(\tilde{B})$. Note that, because the domain of $\tilde{B}$ is Y, $c_l(\tilde{B})$ and $c_l(\tilde{B})$ are located on the y-axis.

After centroid type-reduction, defuzzification of $[c_l(\tilde{B}), c_r(\tilde{B})]$ leads to the defuzzified output of the IT2 Mamdani fuzzy system, $y_c(\mathbf{x}')$, where

$$y_c(\mathbf{x}') = \frac{1}{2}[c_l(\tilde{B}) + c_r(\tilde{B})] \tag{9.105}$$

As in (3.38), $y_c(\mathbf{x}')$ is shown as an explicit function of $\mathbf{x}'$ because $c_l(\tilde{B})$ and $c_l(\tilde{B})$ depend upon $\underline{\mu}_{\tilde{B}}(y_i|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}}(y_i|\mathbf{x}')$ which are functions of the input to the fuzzy system. For each $\mathbf{x}'$ a different value is obtained for $y_c.y_c(\mathbf{x}')$ is usually difficult and time-consuming to compute because of first having to compute the union in (9.95)–(9.97), and therefore it is not widely used in real-world applications of IT2 fuzzy systems.

9.6.2 Height Type-Reduction + Defuzzification for an IT2 Mamdani Fuzzy System

Height type-reduction in an IT2 Mamdani fuzzy system leads to the height type-reduced set, $Y_h(\mathbf{x}')$, where (see Sect. 8.3.3)

$$Y_h(\mathbf{x}') = 1/[y_l^h(\mathbf{x}'),\ y_r^h(\mathbf{x}')] \tag{9.106}$$

In (9.106), $y_l^h(\mathbf{x}')$ and $y_r^h(\mathbf{x}')$ are computed as explained in Theorem 8.2, as:

$$y_l^h(\mathbf{x}') = \frac{\sum_{i=1}^{L} \bar{y}^i \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') + \sum_{i=L+1}^{M} \bar{y}^i \underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')}{\sum_{i=1}^{L} \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') + \sum_{i=L+1}^{M} \underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')} \tag{9.107}$$

$$y_r^h(\mathbf{x}') = \frac{\sum_{i=1}^{R} \bar{y}^i \underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') + \sum_{i=R+1}^{M} \bar{y}^i \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')}{\sum_{i=1}^{R} \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') + \sum_{i=R+1}^{M} \bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')} \tag{9.108}$$

In (9.107) and (9.108), $\bar{y}^i$ is the point having the highest primary membership in the principal MF of the fired-rule output set $\tilde{B}^i$, and $\underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}')$ are computed from Corollaries 9.2, 9.4 or 9.6, in which y is set equal to $\bar{y}^i$. EIASC or EKM algorithms can be used to compute $y_l^h(\mathbf{x}')$ and $y_r^h(\mathbf{x}')$, both of which are located on the y-axis.

Height type-reduction does not begin with an FOU or DOU, which makes it very different from centroid type-reduction.

After height type-reduction, defuzzification of $[y_l^h(\mathbf{x}'),\ y_r^h(\mathbf{x}')]$ leads to the defuzzified output of the IT2 Mamdani fuzzy system, $y_h(\mathbf{x}')$, where

$$y_h(\mathbf{x}') = \frac{1}{2}[y_l^h(\mathbf{x}') + y_r^h(\mathbf{x}')] \tag{9.109}$$

9.6.3 COS Type-Reduction + Defuzzification for an IT2 Mamdani Fuzzy System

COS type-reduction in an IT2 Mamdani fuzzy system leads to the COS type-reduced set, $Y_{\text{COS}}(\mathbf{x}')$, where (see Sect. 8.3.4)

$$Y_{\text{COS}}(\mathbf{x}') = 1/[y_l^{\text{COS}}(\mathbf{x}'),\ y_r^{\text{COS}}(\mathbf{x}')] \tag{9.110}$$

In (9.110), $y_l^{\text{COS}}(\mathbf{x}')$ and $y_r^{\text{COS}}(\mathbf{x}')$ are computed as explained in Theorem 8.3, as:

$$y_l^{\text{COS}}(\mathbf{x}') = \frac{\sum_{i=1}^{L} c_l(\tilde{G}^i)\bar{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M} c_l(\tilde{G}^i)\underline{f}^i(\mathbf{x}')}{\sum_{i=1}^{L} \bar{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M} \underline{f}^i(\mathbf{x}')} \tag{9.111}$$

$$y_r^{\text{COS}}(\mathbf{x}') = \frac{\sum_{i=1}^{R} c_r(\tilde{G}^i)\underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M} c_r(\tilde{G}^i)\bar{f}^i(\mathbf{x}')}{\sum_{i=1}^{R} \underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M} \bar{f}^i(\mathbf{x}')} \tag{9.112}$$

In (9.111) and (9.112), $c_l(\tilde{G}^i)$ and $c_r(\tilde{G}^i)$ are the left- and right-endpoints of the centroid of the IT2 consequent $\tilde{G}^i$, and $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are the endpoints of the firing interval for the ith rule, computed using (9.25) and (9.26) for singleton fuzzification, (9.51) and (9.52) for T1 non-singleton fuzzification, and (9.82) and (9.83) for IT2 non-singleton fuzzification. EIASC or EKM algorithms can be used to compute $y_l^{\text{COS}}(\mathbf{x}')$ and $y_r^{\text{COS}}(\mathbf{x}')$, both of which are located on the y-axis.

The centroid of each rule's IT2 consequent set, $[c_l(\tilde{G}^i),\ c_r(\tilde{G}^i)]$, only has to be computed once (and then stored) after an IT2 Mamdani fuzzy system has been designed, because those centroids do not depend upon the input $\mathbf{x} = \mathbf{x}'$ to the fuzzy system.

Center-of-sets type-reduction also does not begin with an FOU or DOU, which also makes it very different from centroid type-reduction.

After center-of-sets type-reduction, defuzzification of $[y_l^{\text{COS}}(\mathbf{x}'),\ y_r^{\text{COS}}(\mathbf{x}')]$ leads to the defuzzified output of the IT2 Mamdani fuzzy system, $y_{\text{COS}}(\mathbf{x}')$, where

$$y_{\text{COS}}(\mathbf{x}') = \frac{1}{2}[y_l^{\text{COS}}(\mathbf{x}') + y_r^{\text{COS}}(\mathbf{x}')] \tag{9.113}$$

9.6.4 *Type-Reduction + Defuzzification for an IT2 TSK Fuzzy System*

In Sect. 3.6.4 two kinds of defuzzifiers were introduced for a T1 TSK fuzzy system, leading to:

- Unnormalized TSK fuzzy system.
- Normalized TSK fuzzy system.

Additionally, in Example 9.1, two different kinds of consequents were introduced for an IT2 TSK rule:

- A2-C0: Antecedents are IT2 FSs and consequent $g^l = c_0^l + c_1^l x_1 + c_2^l x_2 + \cdots + c_p^l x_p$, in which the c_i^l coefficients are numbers.
- A2-C1: Antecedents are IT2 FSs and consequent $G^l = C_0^l + C_1^l x_1 + C_2^l x_2 + \cdots + C_p^l x_p$, in which the C_i^l coefficients are *type-1 interval fuzzy numbers.*

Consequently (as of the year 2017), there can be four kinds of IT2 TSK fuzzy systems:

1. Unnormalized A2-C0 IT2 TSK fuzzy system.
2. Normalized A2-C0 IT2 TSK fuzzy system.
3. Unnormalized A2-C1 IT2 TSK fuzzy system.
4. Normalized A2-C1 IT2 TSK fuzzy system.

Each of these IT2 TSK fuzzy systems is described in this section.

9.6.4.1 Unnormalized A2-C0 IT2 TSK Fuzzy System

In this simplest of all IT2 TSK fuzzy systems, one begins by computing $Y_{\text{TSK}}^U(\mathbf{x}')$, where

$$Y_{\text{TSK}}^{U}(\mathbf{x}') = 1/[y_{\text{TSK},l}^{U}(\mathbf{x}'),\ y_{\text{TSK},r}^{U}(\mathbf{x}')] \tag{9.114}$$

in which

$$y_{\text{TSK},l}^{U}(\mathbf{x}') = \sum_{i=1}^{M} g^{i}(\mathbf{x}')\underline{f}^{i}(\mathbf{x}') \tag{9.115}$$

$$y_{\text{TSK},r}^{U}(\mathbf{x}') = \sum_{i=1}^{M} g^{i}(\mathbf{x}')\bar{f}^{i}(\mathbf{x}') \tag{9.116}$$

$$g^{i}(\mathbf{x}') = c_0^i + c_1^i x_1' + c_2^i x_2' + \cdots + c_p^i x_p' \tag{9.117}$$

and $\underline{f}^{i}(\mathbf{x}')$ and $\bar{f}^{i}(\mathbf{x}')$ are the endpoints of the firing interval for the ith rule, computed using (9.25) and (9.26) for singleton fuzzification, (9.51) and (9.52) for T1 non-singleton fuzzification, and (9.82) and (9.83) for IT2 non-singleton fuzzification.

After $y_{\text{TSK},l}^{U}(\mathbf{x}')$ and $y_{\text{TSK},r}^{U}(\mathbf{x}')$ have been computed, $y_{\text{TSK}}^{U}(\mathbf{x}')$ is computed as:

$$y_{\text{TSK}}^{U}(\mathbf{x}') = \frac{1}{2}[y_{\text{TSK},l}^{U}(\mathbf{x}') + y_{\text{TSK},r}^{U}(\mathbf{x}')] \tag{9.118}$$

Observe that no iterative computations are needed to compute (9.115) and (9.116), and that (9.118) resembles the output of the T1 TSK fuzzy system in (3.42) when the firing level is the average of the endpoints of the firing interval for the IT2 TSK fuzzy system, i.e.,

$$y_{\text{TSK}}^{U}(\mathbf{x}') = \sum_{i=1}^{M} g^{i}(\mathbf{x}')\ [\underline{f}^{i}(\mathbf{x}') + \bar{f}^{i}(\mathbf{x}')]/2 \tag{9.119}$$

9.6.4.2 Normalized A2-C0 IT2 TSK Fuzzy System

For this IT2 TSK fuzzy system, one begins by computing $Y_{\text{TSK}}(\mathbf{x}')$, where (Liang and Mendel 2001):

$$\begin{aligned} Y_{\text{TSK}}^{N}(\mathbf{x}') &= 1\Big/\Big[y_{\text{TSK},l}^{N}(\mathbf{x}'),\ y_{\text{TSK},r}^{N}(\mathbf{x}')\Big] \\ &= 1\Big/\int_{f^1(\mathbf{x}')\in\left[\underline{f}^1(\mathbf{x}'),\bar{f}^1(\mathbf{x}')\right]}\cdots\int_{f^M(\mathbf{x}')\in\left[\underline{f}^M(\mathbf{x}'),\bar{f}^M(\mathbf{x}')\right]}\left[\frac{\sum_{i=1}^{M} f^i(\mathbf{x}')\,g^i(\mathbf{x}')}{\sum_{i=1}^{M} f^i(\mathbf{x}')}\right] \end{aligned} \tag{9.120}$$

This equation resembles the equation of height type-reduction in (8.29) and (8.30), when $\bar{y}^i \to g^i(\mathbf{x}')$, $\underline{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') \to \underline{f}^i(\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}^i}(\bar{y}^i|\mathbf{x}') \to \bar{f}^i(\mathbf{x}')$; hence, [see (9.107) and (9.108)]

$$y^N_{\mathrm{TSK},l}(\mathbf{x}') = \frac{\sum_{i=1}^{L} g^i(\mathbf{x}')\bar{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M} g^i(\mathbf{x}')\underline{f}^i(\mathbf{x}')}{\sum_{i=1}^{L} \bar{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M} \underline{f}^i(\mathbf{x}')} \tag{9.121}$$

$$y^N_{\mathrm{TSK},r}(\mathbf{x}') = \frac{\sum_{i=1}^{R} g^i(\mathbf{x}')\underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M} g^i(\mathbf{x}')\bar{f}^i(\mathbf{x}')}{\sum_{i=1}^{R} \underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M} \bar{f}^i(\mathbf{x}')} \tag{9.122}$$

In (9.121) and (9.122), $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are the endpoints of the firing interval for the ith rule, computed using (9.25) and (9.26) for singleton fuzzification, (9.51) and (9.52) for T1 non-singleton fuzzification, and (9.82) and (9.83) for IT2 non-singleton fuzzification. EIASC or EKM algorithms can be used to compute $y^N_{\mathrm{TSK},l}(\mathbf{x}')$ and $y^N_{\mathrm{TSK},r}(\mathbf{x}')$, both of which are located on the y-axis.

After $y^N_{\mathrm{TSK},l}(\mathbf{x}')$ and $y^N_{\mathrm{TSK},r}(\mathbf{x}')$ have been computed, $y^N_{\mathrm{TSK}}(\mathbf{x}')$ is computed using (9.118) in which normalized quantities replace the unnormalized ones.

9.6.4.3 Unnormalized A2-C1 IT2 TSK Fuzzy System

To begin, Example 2.22 is applied to the consequent $G^l(\mathbf{x}') = C_0^l + C_1^l x_1' + C_2^l x_2' + \cdots + C_p^l x_p'$, where

$$C_j^l = [l_j^l, r_j^l] \quad j = 1, \ldots, p, \tag{9.123}$$

to conclude that $G^l(\mathbf{x}')$ is a type-1 interval fuzzy number (Definition 2.6) whose domain is $\left[\sum_{j=1}^{p} x_j' l_j^l + l_0^l,\ \sum_{j=1}^{p} x_j' r_j^l + r_0^l\right]$. To simplify notation, let

$$g_l^l(\mathbf{x}') \equiv g_l^l = \sum_{j=1}^{p} x_j' l_j^l + l_0^l \tag{9.124}$$

$$g_r^l(\mathbf{x}') \equiv g_r^l = \sum_{j=1}^{p} x_j' r_j^l + r_0^l \tag{9.125}$$

For this IT2 TSK fuzzy system, one begins by computing $Y^U_{\mathrm{TSK}}(\mathbf{x}')$, where:

$$
\begin{aligned}
Y_{\mathrm{TSK}}^{U}(\mathbf{x}') &= 1\Big/\left[y_{\mathrm{TSK},l}^{U}(\mathbf{x}'),\; y_{\mathrm{TSK},r}^{U}(\mathbf{x}')\right] \\
&= 1/\int_{G^1(\mathbf{x}')\in\left[g_l^1, g_r^1\right]}\cdots\int_{G^M(\mathbf{x}')\in\left[g_l^M,\, g_r^M\right]}\int_{f^1(\mathbf{x}')\in\left[\underline{f}^1(\mathbf{x}'),\bar{f}^1(\mathbf{x}')\right]}\cdots\int_{f^M(\mathbf{x}')\in\left[\underline{f}^M(\mathbf{x}'),\bar{f}^M(\mathbf{x}')\right]}\left[\sum_{i=1}^{M} f^i(\mathbf{x}')G^i(\mathbf{x}')\right]
\end{aligned}
\tag{9.126}
$$

It should be clear by now, that:

$$
y_{\mathrm{TSK},l}^{U}(\mathbf{x}') = \sum_{i=1}^{M} \underline{f}^i(\mathbf{x}')\, g_l^i(\mathbf{x}') \tag{9.127}
$$

$$
y_{\mathrm{TSK},r}^{U}(\mathbf{x}') = \sum_{i=1}^{M} \bar{f}^i(\mathbf{x}')\, g_r^i(\mathbf{x}') \tag{9.128}
$$

In (9.127) and (9.128), $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are the endpoints of the firing interval for the ith rule, computed using (9.25) and (9.26) for singleton fuzzification, (9.51) and (9.52) for T1 non-singleton fuzzification, and (9.82) and (9.83) for IT2 non-singleton fuzzification. Observe that no iterative computations are needed to compute (9.127) and (9.128).

After $y_{\mathrm{TSK},l}^{U}(\mathbf{x}')$ and $y_{\mathrm{TSK},r}^{U}(\mathbf{x}')$ have been computed, $y_{\mathrm{TSK}}^{U}(\mathbf{x}')$ is computed using (9.118).

9.6.4.4 Normalized A2-C1 IT2 TSK Fuzzy System

For this IT2 TSK fuzzy system, (9.124) and (9.125) also apply, and:

$$
\begin{aligned}
Y_{\mathrm{TSK}}^{N}(\mathbf{x}') &= 1/\left[y_{\mathrm{TSK},l}^{N}(\mathbf{x}'),\; y_{\mathrm{TSK},r}^{N}(\mathbf{x}')\right] \\
&= 1/\int_{G^1(\mathbf{x}')\in\left[g_l^1,\, g_r^1\right]}\cdots\int_{G^M(\mathbf{x}')\in\left[g_l^M,\, g_r^M\right]}\int_{f^1(\mathbf{x}')\in\left[\underline{f}^1(\mathbf{x}'),\bar{f}^1(\mathbf{x}')\right]}\cdots\int_{f^M(\mathbf{x}')\in\left[\underline{f}^M(\mathbf{x}'),\bar{f}^M(\mathbf{x}')\right]}\left[\frac{\sum_{i=1}^{M} f^i(\mathbf{x}')G^i(\mathbf{x}')}{\sum_{i=1}^{M} f^i(\mathbf{x}')}\right]
\end{aligned}
\tag{9.129}
$$

Now, analogous to (9.121) and (9.122):

$$
y_{\mathrm{TSK},l}^{N}(\mathbf{x}') = \frac{\sum_{i=1}^{L} g_l^i(\mathbf{x}')\bar{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M} g_l^i(\mathbf{x}')\underline{f}^i(\mathbf{x}')}{\sum_{i=1}^{L}\bar{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M}\underline{f}^i(\mathbf{x}')} \tag{9.130}
$$

$$
y_{\mathrm{TSK},r}^{N}(\mathbf{x}') = \frac{\sum_{i=1}^{R} g_r^i(\mathbf{x}')\underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M} g_r^i(\mathbf{x}')\bar{f}^i(\mathbf{x}')}{\sum_{i=1}^{R}\underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M}\bar{f}^i(\mathbf{x}')} \tag{9.131}
$$

In (9.130) and (9.131), $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are the endpoints of the firing interval for the ith rule, computed using (9.25) and (9.26) for singleton fuzzification, (9.51) and

(9.52) for T1 non-singleton fuzzification, and (9.82) and (9.83) for IT2 non-singleton fuzzification. EIASC or EKM algorithms can be used to compute $y_{\text{TSK},l}^{N}(\mathbf{x}')$ and $y_{\text{TSK},r}^{N}(\mathbf{x}')$, both of which are located on the y-axis.

After $y_{\text{TSK},l}^{N}(\mathbf{x}')$ and $y_{\text{TSK},r}^{N}(\mathbf{x}')$ have been computed, $y_{\text{TSK}}^{N}(\mathbf{x}')$ is computed using (9.118) in which normalized quantities replace the unnormalized ones.

9.6.4.5 Remarks

When (9.121) and (9.122), or (9.130) and (9.131) are compared with (9.111) and (9.112), it is clear that there is a difference between the IT2 TSK and Mamdani fuzzy systems only when the IT2 TSK rule consequents are not constants. This is consistent with the same observation that was made about type-1 TSK and Mamdani fuzzy systems.

Unnormalized IT2 TSK fuzzy systems do not need EIASC or EKM iterative algorithms, which make them attractive for real-time systems. Interestingly, there are also normalized IT2 TSK fuzzy systems that do not need EIASC or EKM iterative algorithms, as is demonstrated in Sect. 9.9.2. In an IT2 FLC, where control is usually limited, using a normalized IT2 TSK fuzzy system is arguably better than using an unnormalized IT2 TSK fuzzy system.

9.6.5 Novelty Partitions

Recall that IT2 second-order rule partitions (Definition 9.9) focus on when the LMF or the UMF of an FOU of an IT2 FS that is associated with either x_1, or x_2, or …, or x_p changes its mathematical formula within an IT2 first-order rule partition. Such changes produce changes in the firing interval within an IT2 first-order rule partition.

Type-reduction also makes use of the LMF or the UMF of an FOU of an IT2 FS that is associated with either x_1, or x_2, or …, or x_p within an IT2 first-order rule partition, and it does this in a *novel* way (Wu 2011, 2012, 2013b) (see, also Sect. 9.13.2). Type-reduction can be interpreted as a further partitioning of IT2 first-order rule partitions into what can be called *IT2 novelty rule partitions*.

Definition 9.10 *IT2 novelty rule partitions* of $X_1 \times X_2 \times \cdots \times X_p$ occur only when type-reduction is used, and result from different endpoints of a firing interval being used to compute the endpoints of the type-reduced set. It occurs within an IT2 first-order rule partition, regardless of whether or not there are any IT2 second-order rule partitions.

Unfortunately, it is very difficult to determine and display the geometry of IT2 novelty rule partitions because there are no closed-form formulas for the endpoints of a type-reduced set. This does not diminish the importance of IT2 novelty rule partitions, because they provide us with additional insight into the further partitioning of $X_1 \times X_2 \times \cdots \times X_p$, something that can only occur for an IT2 fuzzy system that uses type-reduction, and can never occur in a T1 fuzzy system. IT2 novelty rule partitions may help to explain why system performance is often arguably better for an IT2 fuzzy system that uses type-reduction than it is for one that does not use type-reduction (e.g., see Sect. 10.8.3).

Some examples of IT2 novelty rule partitions are in Mendel et al. (2014, Chaps. 4 and 5). See, also, the discussion that is given in Sect. 9.13.2 about novelty.

9.7 Comprehensive Example

This example is an IT2 version of the comprehensive example that is in Sect. 3.7. Consider[9] an IT2 fuzzy system with two inputs (x_1 and x_2) and one output (u). Each input domain consists of two shoulder IT2 FSs, shown as the shaded areas in Fig. 9.13. The rule-base consists of the following four IT2 Zadeh rules:

$$\begin{array}{l} \tilde{R}_Z^1: \text{ IF } x_1 \text{ is } \tilde{F}_1^1 = \tilde{X}_{11} \text{ and } x_2 \text{ is } \tilde{F}_2^1 = \tilde{X}_{21}, \text{ THEN } y \text{ is } \tilde{G}^1 \\ \tilde{R}_Z^2: \text{ IF } x_1 \text{ is } \tilde{F}_1^2 = \tilde{X}_{11} \text{ and } x_2 \text{ is } \tilde{F}_2^2 = \tilde{X}_{22}, \text{ THEN } y \text{ is } \tilde{G}^2 \\ \tilde{R}_Z^3: \text{ IF } x_1 \text{ is } \tilde{F}_1^3 = \tilde{X}_{12} \text{ and } x_2 \text{ is } \tilde{F}_2^3 = \tilde{X}_{21}, \text{ THEN } y \text{ is } \tilde{G}^3 \\ \tilde{R}_Z^4: \text{ IF } x_1 \text{ is } \tilde{F}_1^4 = \tilde{X}_{12} \text{ and } x_2 \text{ is } \tilde{F}_2^4 = \tilde{X}_{22}, \text{ THEN } y \text{ is } \tilde{G}^4 \end{array} \tag{9.132}$$

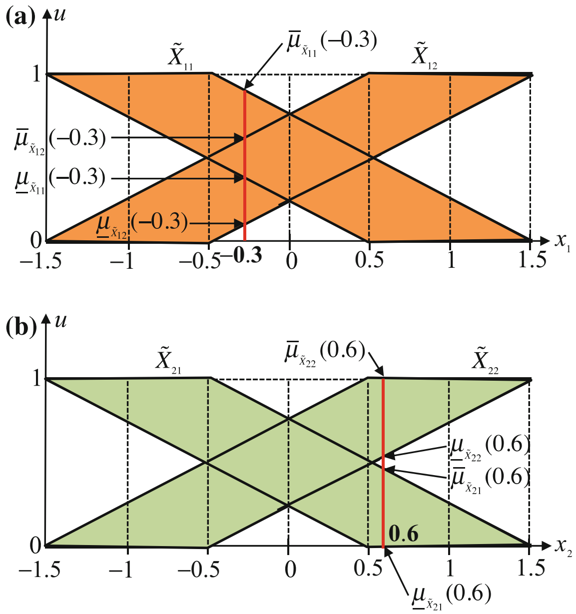


Fig. 9.13 FOUs of the IT2 FSs. **a** FOUs for x_1; and **b** FOUs for x_2. The measured values of x_1 ($x_1' = -0.3$) and ($x_2' = 0.6$) are shown in *bold face*

[9]Much of this example was provided by Dongrui Wu, and is used with his permission.

Table 9.1 Rule-base and consequent centroids of the IT2 fuzzy system[a]

x_1	x_2	
	$\tilde{X}_{21}$	$\tilde{X}_{22}$
$\tilde{X}_{11}$	$\tilde{R}_Z^1 : C_{\tilde{G}^1} = [a^1, b^1] \equiv [-1, -0.9]$	$\tilde{R}_Z^2 : C_{\tilde{G}^2} = [a^2, b^2] \equiv [-0.6, -0.4]$
$\tilde{X}_{12}$	$\tilde{R}_Z^3 : C_{\tilde{G}^3} = [a^3, b^3] \equiv [0.4, 0.6]$	$\tilde{R}_Z^4 : C_{\tilde{G}^4} = [a^4, b^4] \equiv [0.9, 1]$

[a] a^i and b^i are short for $c_l(\tilde{G}^i)$ and $c_r(\tilde{G}^i)$, respectively

The centroids of the corresponding rule consequents are given in Table 9.1; their numerical values were chosen arbitrarily ahead of time. Comparing the centroids in Table 9.1 with the COGs in Table 3.1, observe that the latter are contained within the former.

Here $\mathbf{x}' = \mathrm{col}(-0.3, 0.6)$, whose components are shown bold-faced in Fig. 9.13. From that figure [or by writing the equations for the sloped portions of the lower and upper MFs and then evaluating them at $\mathbf{x}' = \mathrm{col}(-0.3, 0.6)$] the following firing intervals are obtained for each of the IT2 FSs:

$$\begin{cases} [\underline{\mu}_{\tilde{X}_{11}}(-0.3),\ \bar{\mu}_{\tilde{X}_{11}}(-0.3)] = [0.4,\ 0.9] \\ [\underline{\mu}_{\tilde{X}_{12}}(-0.3),\ \bar{\mu}_{\tilde{X}_{12}}(-0.3)] = [0.1,\ 0.6] \\ [\underline{\mu}_{\tilde{X}_{21}}(0.6),\ \bar{\mu}_{\tilde{X}_{21}}(0.6)] = [0,\ 0.45] \\ [\underline{\mu}_{\tilde{X}_{22}}(0.6),\ \bar{\mu}_{\tilde{X}_{22}}(0.6)] = [0.55,\ 1] \end{cases} \tag{9.133}$$

The firing intervals for the four rules, computed using the product t-norm, and singleton fuzzification, are shown in Table 9.2. Comparing the firing intervals in Table 9.2 with the firing levels in Table 3.2, observe that the latter are contained within the former.

Using the EIASC or EKM algorithms and COS type-reduction, it is found that $L = 1$ and $R = 3$; hence, (9.111) and (9.112) are:

$$\begin{aligned} y_l^{\mathrm{COS}}(\mathbf{x}') &= \frac{\bar{f}^1 a^1 + \underline{f}^2 a^2 + \underline{f}^3 a^3 + \underline{f}^4 a^4}{\bar{f}^1 + \underline{f}^2 + \underline{f}^3 + \underline{f}^4} \\ &= \frac{0.405 \times (-1) + 0.22 \times (-0.6) + 0 \times 0.4 + 0.055 \times 0.9}{0.405 + 0.22 + 0 + 0.055} \\ &= -0.717 \end{aligned} \tag{9.134}$$

$$\begin{aligned} y_r^{\mathrm{COS}}(\mathbf{x}') &= \frac{\underline{f}^1 b^1 + \underline{f}^2 b^2 + \underline{f}^3 b^3 + \bar{f}^4 b^4}{\underline{f}^1 + \underline{f}^2 + \underline{f}^3 + \bar{f}^4} \\ &= \frac{0 \times (-0.9) + 0.22 \times (-0.4) + 0 \times 0.6 + 0.6 \times 1}{0 + 0.22 + 0 + 0.6} \\ &= 0.624 \end{aligned} \tag{9.135}$$

Table 9.2 Firing intervals computed for $\mathbf{x}' = \mathrm{col}(-0.3, 0.6)$ using (9.25) and (9.26), and rule consequent centroid

Rule number	Firing interval	Rule consequent centroid[a]
$\tilde{R}_Z^1$	$[\underline{f}^1, \bar{f}^1] = [\underline{\mu}_{\tilde{X}_{11}}(-0.3) \times \underline{\mu}_{\tilde{X}_{21}}(0.6),\ \bar{\mu}_{\tilde{X}_{11}}(-0.3) \times \bar{\mu}_{\tilde{X}_{21}}(0.6)]$	$C_{\tilde{G}_1} = [a^1, b^1] \equiv [-1, -0.9]$
	$= [0.4 \times 0, 0.9 \times 0.45] = [0, 0.405]$	
$\tilde{R}_Z^2$	$[\underline{f}^2, \bar{f}^2] = [\underline{\mu}_{\tilde{X}_{11}}(-0.3) \times \underline{\mu}_{\tilde{X}_{22}}(0.6),\ \bar{\mu}_{\tilde{X}_{11}}(-0.3) \times \bar{\mu}_{\tilde{X}_{22}}(0.6)]$	$C_{\tilde{G}_2} \equiv [a^2, b^2] = [-0.6, -0.4]$
	$= [0.4 \times 0.55, 0.9 \times 1] = [0.22, 0.9]$	
$\tilde{R}_Z^3$	$[\underline{f}^3, \bar{f}^3] = [\underline{\mu}_{\tilde{X}_{12}}(-0.3) \times \underline{\mu}_{\tilde{X}_{21}}(0.6),\ \bar{\mu}_{\tilde{X}_{12}}(-0.3) \times \bar{\mu}_{\tilde{X}_{21}}(0.6)]$	$C_{\tilde{G}_3} \equiv [a^3, b^3] = [0.4, 0.6]$
	$= [0.1 \times 0, 0.6 \times 0.45] = [0, 0.27]$	
$\tilde{R}_Z^4$	$[\underline{f}^4, \bar{f}^4] = [\underline{\mu}_{\tilde{X}_{12}}(-0.3) \times \underline{\mu}_{\tilde{X}_{22}}(0.6),\ \bar{\mu}_{\tilde{X}_{12}}(-0.3) \times \bar{\mu}_{\tilde{X}_{22}}(0.6)]$	$C_{\tilde{G}_4} \equiv [a^4, b^4] = [0.9, 1]$
	$= [0.1 \times 0.55, 0.6 \times 1] = [0.055, 0.6]$	

[a] a^i and b^i are short for $c_l(\tilde{G}^i)$ and $c_r(\tilde{G}^i)$, respectively. Observe that $a_1 < a_2 < a_3 < a_4$ and $b_1 < b_2 < b_3 < b_4$, so no rule-reordering is needed to use EIASC or EKM algorithms

Finally, the defuzzified output in (9.113) is

$$y_{\cos}(\mathbf{x}') = (-0.717 + 0.624)/2 = -0.047 \tag{9.136}$$

Next, the Zadeh rules are changed to the following A2-C0 IT2 TSK rules:

$$\begin{aligned}
\tilde{R}^1_{\mathrm{TSK}} &: \text{ IF } x_1 \text{ is } \tilde{F}^1_1 = \tilde{X}_{11} \text{ and } x_2 \text{ is } \tilde{F}^1_2 = \tilde{X}_{21}, \text{ THEN } y \text{ is } g^1(\mathbf{x}) = c^1_0 + c^1_1 x_1 + c^1_2 x_2 \\
\tilde{R}^2_{\mathrm{TSK}} &: \text{ IF } x_1 \text{ is } \tilde{F}^2_1 = \tilde{X}_{11} \text{ and } x_2 \text{ is } \tilde{F}^2_2 = \tilde{X}_{22}, \text{ THEN } y \text{ is } g^2(\mathbf{x}) = c^2_0 + c^2_1 x_1 + c^2_2 x_2 \\
\tilde{R}^3_{\mathrm{TSK}} &: \text{ IF } x_1 \text{ is } \tilde{F}^3_1 = \tilde{X}_{12} \text{ and } x_2 \text{ is } \tilde{F}^3_2 = \tilde{X}_{21}, \text{ THEN } y \text{ is } g^3(\mathbf{x}) = c^3_0 + c^3_1 x_1 + c^3_2 x_2 \\
\tilde{R}^4_{\mathrm{TSK}} &: \text{ IF } x_1 \text{ is } \tilde{F}^4_1 = \tilde{X}_{12} \text{ and } x_2 \text{ is } \tilde{F}^4_2 = \tilde{X}_{22}, \text{ THEN } y \text{ is } g^4(\mathbf{x}) = c^4_0 + c^4_1 x_1 + c^4_2 x_2
\end{aligned} \tag{9.137}$$

The firing intervals for the four rules, as well as their consequents, are shown in Table 9.3 for $\mathbf{x}' = \mathrm{col}(-0.3, 0.6)$, where, as in Sect. 3.7, $c^1_0 = 1$, $c^2_0 = 2$, $c^3_0 = 3$, $c^4_0 = 4$, $c^1_1 = 1.5$, $c^2_1 = 2.5$, $c^3_1 = 3.5$, $c^4_1 = 4.5$, $c^1_2 = 2$, $c^2_2 = 2.5$, $c^3_2 = 3$, and $c^4_2 = 3.5$.

For the *unnormalized* A2-C0 IT2 TSK fuzzy system, $y^U_{\mathrm{TSK}}(\mathbf{x}')$ in (9.119) is:

$$\begin{aligned}
y^U_{\mathrm{TSK}}(\mathbf{x}') &= \frac{1}{2}\sum_{i=1}^{4} g^i(\mathbf{x}')\,[\underline{f}^i(\mathbf{x}') + \bar{f}^i(\mathbf{x}')] \\
&= \frac{1}{2}[1.75(0 + 0.405) + 2.75(0.22 + 0.9) + 3.75(0 + 0.27) + 4.75(0.055 + 0.6)] \\
&= 3.96
\end{aligned} \tag{9.138}$$

For the *normalized* A2-C0 IT2 TSK fuzzy system, applying the EIASC or EKM algorithms to $y^N_{\mathrm{TSK},l}(\mathbf{x}')$ and $y^N_{\mathrm{TSK},r}(\mathbf{x}')$, in (9.121) and (9.122), respectively (observe that g^1, g^2, g^3 and g^4 are already rank-ordered, so no reordering is needed for this example), one finds $L = 1$ and $R = 3$; hence,

$$\begin{aligned}
y^N_{\mathrm{TSK},l}(\mathbf{x}') &= \frac{\bar{f}^1 g^1 + \underline{f}^2 g^2 + \underline{f}^3 g^3 + \underline{f}^4 g^4}{\bar{f}^1 + \underline{f}^2 + \underline{f}^3 + \underline{f}^4} \\
&= \frac{0.405 \times (1.75) + 0.22 \times (2.75) + 0 \times 3.75 + 0.055 \times 4.75}{0.405 + 0.22 + 0 + 0.055} \\
&= 2.32
\end{aligned} \tag{9.139}$$

$$\begin{aligned}
y^N_{\mathrm{TSK},r}(\mathbf{x}') &= \frac{\underline{f}^1 g^1 + \underline{f}^2 g^2 + \underline{f}^3 g^3 + \bar{f}^4 g^4}{\underline{f}^1 + \underline{f}^2 + \underline{f}^3 + \bar{f}^4} \\
&= \frac{0 \times (1.75) + 0.22 \times (2.75) + 0 \times 3.75 + 0.6 \times 4.75}{0 + 0.22 + 0 + 0.6} \\
&= 4.21
\end{aligned} \tag{9.140}$$

Finally, the defuzzified output is

Table 9.3 Firing intervals computed for $\mathbf{x}' = \mathrm{col}(-0.3, 0.6)$ using (9.25) and (9.26) and the A2-C0 TSK rule consequents

Rule number	Firing interval	Rule consequent
$\tilde{R}^1_{\mathrm{TSK}}$	$[\underline{f}^1, \bar{f}^1] = [\underline{\mu}_{\tilde{X}_{11}}(-0.3) \times \underline{\mu}_{\tilde{X}_{21}}(0.6),\ \bar{\mu}_{\tilde{X}_{11}}(-0.3) \times \bar{\mu}_{\tilde{X}_{21}}(0.6)]$ $= [0.4 \times 0,\ 0.9 \times 0.45] = [0,\ 0.405]$	$g^1(\mathbf{x}') = c_0^1 - 0.3c_1^1 + 0.6c_2^1$ $= 1.75$
$\tilde{R}^2_{\mathrm{TSK}}$	$[\underline{f}^2, \bar{f}^2] = [\underline{\mu}_{\tilde{X}_{11}}(-0.3) \times \underline{\mu}_{\tilde{X}_{22}}(0.6),\ \bar{\mu}_{\tilde{X}_{11}}(-0.3) \times \bar{\mu}_{\tilde{X}_{22}}(0.6)]$ $= [0.4 \times 0.55,\ 0.9 \times 1] = [0.22,\ 0.9]$	$g^2(\mathbf{x}') = c_0^2 - 0.3c_1^2 + 0.6c_2^2$ $= 2.75$
$\tilde{R}^3_{\mathrm{TSK}}$	$[\underline{f}^3, \bar{f}^3] = [\underline{\mu}_{\tilde{X}_{12}}(-0.3) \times \underline{\mu}_{\tilde{X}_{21}}(0.6),\ \bar{\mu}_{\tilde{X}_{12}}(-0.3) \times \bar{\mu}_{\tilde{X}_{21}}(0.6)]$ $= [0.1 \times 0,\ 0.6 \times 0.45] = [0,\ 0.27]$	$g^3(\mathbf{x}') = c_0^3 - 0.3c_1^3 + 0.6c_2^3$ $= 3.75$
$\tilde{R}^4_{\mathrm{TSK}}$	$[\underline{f}^4, \bar{f}^4] = [\underline{\mu}_{\tilde{X}_{12}}(-0.3) \times \underline{\mu}_{\tilde{X}_{22}}(0.6),\ \bar{\mu}_{\tilde{X}_{12}}(-0.3) \times \bar{\mu}_{\tilde{X}_{22}}(0.6)]$ $= [0.1 \times 0.55,\ 0.6 \times 1] = [0.055,\ 0.6]$	$g^4(\mathbf{x}') = c_0^4 - 0.3c_1^4 + 0.6c_2^4$ $= 4.75$

$$y_{\mathrm{TSK}}^{N}(\mathbf{x}') = (2.32 + 4.21)/2 = 3.265 \tag{9.141}$$

The numerical values for $y_{\mathrm{COS}}(\mathbf{x}')$ in (9.136), $y_{\mathrm{TSK}}^{U}(\mathbf{x}')$ in (9.138) and $y_{\mathrm{TSK}}^{N}(\mathbf{x}')$ in (9.141) are all different, and so the reader may be wondering at this point "Which one is the best value?" The reader is reminded that the same FOUs and consequent parameters have been used for all three calculations. In practice, during the designs of each system, its FOU parameters as well as it consequent parameters would be optimized so as to achieve (e.g., minimize) one or more application-specific performance metrics. It is only then that it is meaningful to compare the resulting (9.136), (9.138) and (9.141).

9.8 Approximate Type-Reduction + Defuzzification (Wu–Mendel Uncertainty Bounds) for IT2 Mamdani Fuzzy Systems

Before bypassing type-reduction completely, and going to direct defuzzification, it is worth noting that, for IT2 Mamdani fuzzy systems, there is a middle-of-the-road approach that can be taken, one that is called (by this author) "approximate type-reduction + defuzzification." It obtains a mathematically optimal and formulaic approximation to the type-reduced set, does not require any iterative computations, and is applicable to the three kinds of type-reduction that have been studied, namely centroid, height and center-of-sets; hence, it is very general. Because the derivation of the approximate type-reduced set is complicated and its formulas are too complicated to be used in other kinds of mathematical analyses, such as stability and robustness, its results are only stated. Derivations can be found in Wu and Mendel (2002) and Mendel et al. (2014, Appendix 3A).

The focus here is initially on approximate COS type-reduction. How to use those results for height or centroid type-reduction is explained later.

To begin, four centroids (also called *boundary type-1 fuzzy systems*) are defined, all of which can be computed once the left and right endpoints of the firing interval $(i = 1, \ldots, M)$, $\underline{f}^{i}(\mathbf{x}')$ and $\overline{f}^{i}(\mathbf{x}')$, have been computed. In these centroids, a^{s} and b^{s} are the left- and right-endpoints of the centroid of the sth consequent IT2 FS which are re-ordered in ascending order [of course, the associated variables such as $\underline{f}^{s}(\mathbf{x}')$ and $\overline{f}^{s}(\mathbf{x}')$ need to be relabeled so they remain associated with the correct rule s]. These consequent centroids only have to be computed (and stored) one time after the IT2 Mamdani fuzzy system has been designed, since they do not depend upon the input to the fuzzy system. The boundary T1 fuzzy system centroids are[10]:

---

[10]Note, e.g., that in (9.142) {LMFs, left} refers to the fact that this centroid only uses lower MFs of the firing interval and left-endpoint values of the consequent set centroid. In addition, in (9.142)–(9.145), (9.149) and (9.150), $\underline{f}^{s}(\mathbf{x})$ and $\overline{f}^{s}(\mathbf{x})$ have been shortened to $\underline{f}^{s}$ and $\overline{f}^{s}$, respectively.

$$\{\text{LMFs, left}\}: \; \mathrm{y}_l^{(0)}(\mathbf{x}') = \sum_{s=1}^{M} \underline{f}^s a^s \Big/ \sum_{s=1}^{M} \underline{f}^s \tag{9.142}$$

$$\{\text{LMFs, right}\}: \; y_r^{(M)}(\mathbf{x}') = \sum_{s=1}^{M} \underline{f}^s b^s \Big/ \sum_{s=1}^{M} \underline{f}^s \tag{9.143}$$

$$\{\text{UMFs, left}\}: \; y_l^{(M)}(\mathbf{x}') = \sum_{s=1}^{M} \bar{f}^s a^s \Big/ \sum_{s=1}^{M} \bar{f}^s \tag{9.144}$$

$$\{\text{UMFs, right}\}: \; y_r^{(0)}(\mathbf{x}) = \sum_{s=1}^{M} \bar{f}^s b^s \Big/ \sum_{s=1}^{M} \bar{f}^s \tag{9.145}$$

Theorem 9.3 (Wu–Mendel Uncertainty Bounds—WM UBs) (Wu and Mendel 2002). *The endpoints* $y_l(\mathbf{x}')$ *and* $y_r(\mathbf{x}')$ *of the COS type-reduced set of an IT2 Mamdani fuzzy system for input* $\mathbf{x}'$, *are bounded from the left and right in a mini–max sense, as*:

$$\begin{cases} \underline{y}_l(\mathbf{x}') \le y_l(\mathbf{x}') \le \bar{y}_l(\mathbf{x}') \\ \underline{y}_r(\mathbf{x}') \le y_r(\mathbf{x}') \le \bar{y}_r(\mathbf{x}') \end{cases} \tag{9.146}$$

where:

$$\bar{y}_l(\mathbf{x}') = \min\left\{ y_l^{(0)}(\mathbf{x}'),\; y_l^{(M)}(\mathbf{x}') \right\} \tag{9.147}$$

$$\underline{y}_r(\mathbf{x}') = \max\left\{ y_r^{(0)}(\mathbf{x}'),\; y_r^{(M)}(\mathbf{x}') \right\} \tag{9.148}$$

$$\underline{y}_l(\mathbf{x}') = \bar{y}_l(\mathbf{x}') - \left[\frac{\sum_{s=1}^{M} (\bar{f}^s - \underline{f}^s)}{\sum_{s=1}^{M} \bar{f}^s \sum_{s=1}^{M} \underline{f}^s} \times \frac{\sum_{s=1}^{M} \underline{f}^s (a^s - a^1) \sum_{s=1}^{M} \bar{f}^s (a^M - a^s)}{\sum_{s=1}^{M} \underline{f}^s (a^s - a^1) + \sum_{s=1}^{M} \bar{f}^s (a^M - a^s)} \right] \tag{9.149}$$

$$\bar{y}_r(\mathbf{x}') = \underline{y}_r(\mathbf{x}') + \left[\frac{\sum_{s=1}^{M} (\bar{f}^s - \underline{f}^s)}{\sum_{s=1}^{M} \bar{f}^s \sum_{s=1}^{M} \underline{f}^s} \times \frac{\sum_{s=1}^{M} \bar{f}^s (b^s - b^1) \sum_{s=1}^{M} \underline{f}^s (b^M - b^s)}{\sum_{s=1}^{M} \bar{f}^s (b^s - b^1) + \sum_{s=1}^{M} \underline{f}^s (b^M - b^s)} \right] \tag{9.150}$$

Observe that the four bounds in (9.147)–(9.150) can be computed without having to perform type-reduction. Using these bounds, Wu and Mendel approximate the type-reduced set, as

$$[y_l(\mathbf{x}'), y_r(\mathbf{x}')] \approx [\hat{y}_l(\mathbf{x}'), \hat{y}_r(\mathbf{x}')] = \left[\frac{\underline{y}_l(\mathbf{x}') + \bar{y}_l(\mathbf{x}')}{2}, \frac{\underline{y}_r(\mathbf{x}') + \bar{y}_r(\mathbf{x}')}{2}\right] \tag{9.151}$$

and compute the output of the IT2 fuzzy system, as

$$y_{\mathrm{WMUB}}(\mathbf{x}') = \frac{1}{2}[\hat{y}_l(\mathbf{x}') + \hat{y}_r(\mathbf{x}')] = \frac{1}{2}\left[\frac{\underline{y}_l(\mathbf{x}') + \bar{y}_l(\mathbf{x}')}{2} + \frac{\underline{y}_r(\mathbf{x}') + \bar{y}_r(\mathbf{x}')}{2}\right] \tag{9.152}$$

(instead of as $[y_l(\mathbf{x}') + y_r(\mathbf{x}')]/2$). So, *by using the WM UBs, both an approximate type-reduced set and a defuzzified output* are obtained. When all IT2 MF uncertainties disappear, then (9.152) reduces to a correct type-1 defuzzification formula (Exercise 9.14). Wu and Mendel (2002) also provides the following upper bound on the difference $\delta(\mathbf{x})$ between the defuzzified outputs of the actual type-reduced set and its approximation:

$$\begin{aligned} \delta(\mathbf{x}') &\equiv y(\mathbf{x}') - y_{\mathrm{WMUB}}(\mathbf{x}') \\ &= \left| \left[\frac{y_l(\mathbf{x}') + y_r(\mathbf{x}')}{2}\right] - \frac{1}{2}\left[\frac{\underline{y}_l(\mathbf{x}') + \bar{y}_l(\mathbf{x}')}{2} + \frac{\underline{y}_r(\mathbf{x}') + \bar{y}_r(\mathbf{x}')}{2}\right] \right| \end{aligned} \tag{9.153}$$

$$\delta(\mathbf{x}') \le \frac{1}{4}\left[\left(\bar{y}_l(\mathbf{x}') - \underline{y}_l(\mathbf{x}')\right) + \left(\bar{y}_r(\mathbf{x}') - \underline{y}_r(\mathbf{x}')\right)\right] \tag{9.154}$$

Lynch et al. (2006) replace all of the Mamdani IT2 fuzzy system computations (that include COS type-reduction) with those in (9.142)–(9.150) and (9.152), i.e., (9.142)–(9.150) and (9.152) are their IT2 fuzzy system. This approach is summarized in Fig. 9.14.

How to use the results that are in Theorem 9.3 for centroid and height type-reduction is summarized in Table 9.4, which is self-explanatory.

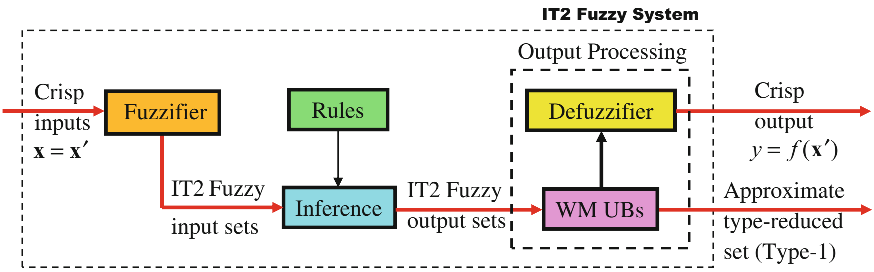


Fig. 9.14 IT2 fuzzy system in which COS type-reduction has been replaced by the WM UBs (Mendel et al. 2014, p. 108, © 2014 Wiley)

Table 9.4 Meanings of a^s, b^s, $\underline{f}^s$, $\bar{f}^s$ and M in the WM UBs for different type-reduction methods

Type-reduction method	a^s and b^s defined	$\underline{f}^s$ and $\bar{f}^s$ defined	M defined
COS	Left and right endpoints of the centroid of the consequent of the sth rule	Lower (left) and upper (right) endpoints of the firing interval of the sth rule	Number of rules
Centroid[a]	$a^s = b^s = y^s$, the sth point in the sampled universe of discourse of the aggregated IT2 fuzzy system's output	Lower and upper MF grades of the sth sample point of the IT2 fuzzy system's output	Number of sampled points
Height	$a^s = b^s = \bar{y}^s$, a single point in the consequent domain of the sth rule (see Sect. 9.6.2)	Lower and upper firing degrees of the sth rule (see Sect. 9.6.2)	Number of rules

[a]Prior to calculating the centroid type-reduced set, the fired-rule output sets must be unioned

9.9 Direct Defuzzification

Referring to Fig. 9.1b, direct defuzzification means not having to perform type-reduction. The ground-rule for direct defuzzification is still the fundamental design requirement: *If all sources of MF uncertainty disappear then a T2 fuzzy system must reduce to a T1 fuzzy system*. Type-reduction + defuzzification is not the only way to satisfy this design requirement. Direct defuzzification can also do this.

Recall, that the main motivation for direct defuzzification is to avoid the iterative computations needed to perform type-reduction. Another motivation for direct defuzzification is to obtain a formula for the crisp output of an IT2 fuzzy system that can be used in some sort of mathematical analyses, e.g., stability, robustness. The direct defuzzification formulas described below not only avoid the iterative computations needed to perform type-reduction, but they are also simple enough to be used in mathematical analyses.

Although only the two simplest direct defuzzification methods are presented below, there are many others, e.g.,

- Wu and Tan (2004, 2005) have a closed-form method that makes use of equivalent type-1 membership grades.
- Coupland and John (2007) have a geometric method that is based on computational geometry.
- Greenfield et al. (2008) have a collapsing method in which each IT2 FS is replaced by a representative embedded T1 FS whose membership grades are computed recursively (see Sect. 9.13.7 for some discussions about this method, because of its connection to Sect. 9.9.1 Nie–Tan method).
- Du and Ying (2010) have an average defuzzifier that first computes crisp outputs obtained for all possible combinations of the lower and upper firing levels, after which the final defuzzified output is computed as the average of all of those crisp outputs.

- Tao et al. (2012) create a linear combination of the outputs of two type-1 fuzzy systems; one uses the possible left-most embedded T1 FSs, and the other uses the possible right-most embedded T1 FSs [how to obtain such embedded sets for IT2 FSs with arbitrary FOUs is not explained, although a construction method for doing this is given pictorially in Wu (2013a, Fig. 14)].
- Juang and Wang (2015) have a method (proposed first in Juang and Juang 2013) that computes the defuzzified outputs as the average of $\bar{y}_l(\mathbf{x}')$ and $\underline{y}_r(\mathbf{x}')$ that are given in (9.147) and (9.148), respectively.

9.9.1 Nie–Tan (NT) Direct Defuzzification

When one begins with the combined fired-rule output set $\tilde{B}$, in (9.95), then (Nie and Tan 2008) defuzzify $\tilde{B}$ directly by computing the COG of the average of its lower and upper MFs, i.e.,

$$y_{\mathrm{NT}}(\mathbf{x}') = \mathrm{COG}\left\{\frac{1}{2}\left[\underline{\mu}_{\tilde{B}}(y|\mathbf{x}') + \bar{\mu}_{\tilde{B}}(y|\mathbf{x}')\right]\right\} = \frac{\sum_{i=1}^{N} y_i\left[\underline{\mu}_{\tilde{B}}(y_i|\mathbf{x}') + \bar{\mu}_{\tilde{B}}(y_i|\mathbf{x}')\right]}{\sum_{i=1}^{N}\left[\underline{\mu}_{\tilde{B}}(y_i|\mathbf{x}') + \bar{\mu}_{\tilde{B}}(y_i|\mathbf{x}')\right]} \tag{9.155}$$

If formulas are available for the lower and upper MFs of $\tilde{B}$, then

$$y_{\mathrm{NT}}(\mathbf{x}') = \frac{\int_a^b y\left[\underline{\mu}_{\tilde{B}}(y|\mathbf{x}') + \bar{\mu}_{\tilde{B}}(y|\mathbf{x}')\right]\mathrm{d}y}{\int_a^b \left[\underline{\mu}_{\tilde{B}}(y|\mathbf{x}') + \bar{\mu}_{\tilde{B}}(y|\mathbf{x}')\right]\mathrm{d}y} \tag{9.156}$$

where $[a, b]$ is the support of $\bar{\mu}_{\tilde{B}}(y|\mathbf{x}')$.

Mendel and Liu (2013) prove that $y_{\mathrm{NT}}(\mathbf{x}')$ in (9.156) is a first-order Taylor series approximation to the actual defuzzified value of $\tilde{B}$, namely $[c_l(\tilde{B}) + c_r(\tilde{B})]/2$, where $c_l(\tilde{B})$ and $c_r(\tilde{B})$ are given in (9.103) and (9.104), respectively. They also have a formula for a correction to $y_{\mathrm{NT}}(\mathbf{x}')$, which when added to $y_{\mathrm{NT}}(\mathbf{x}')$ provides a third-order Taylor series approximation to $[c_l(\tilde{B}) + c_r(\tilde{B})]/2$ (the second-order term is zero). Their examples show that $y_{\mathrm{NT}}(\mathbf{x}')$ is a very good approximation of $[c_l(\tilde{B}) + c_r(\tilde{B})]/2$.

Example 9.12 (Mendel and Liu 2013), Fig. 9.15 depicts the two FOUs from Figs. 8.14 and 8.12 for $\tilde{F}$ and $\tilde{G}$, respectively. Formulas for their lower and upper MFs are in (8.53), (8.54), (8.49) and (8.50). These FOUs are representative of the FOUs that are obtained in an IT2 fuzzy system when two rules are fired and their

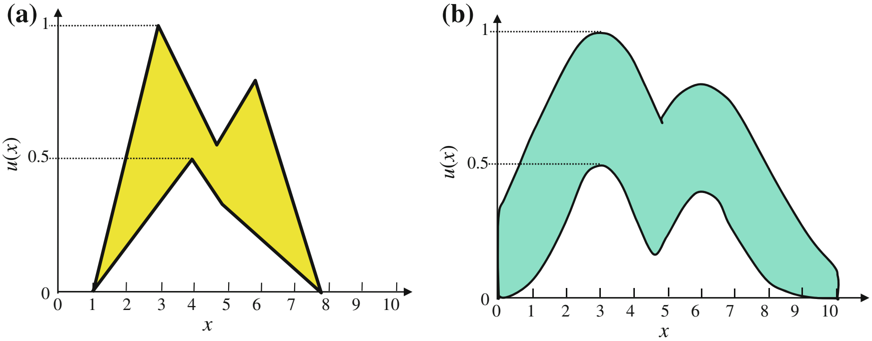


Fig. 9.15 FOUs for Example 9.12. **a** $\tilde{F}$ from Fig. 8.14 and **b** $\tilde{G}$ from Fig. 8.12

fired-rule output sets are combined using the maximum union operation. Using the continuous EKM algorithms in Table 8.14, and $y_{NT}(\mathbf{x}')$ in (9.156), it was found that (to two significant figures):

- $C_{\tilde{F}} = 1/[3.66, 4.99] \rightarrow y_c(\mathbf{x}') = 4.33 \quad y_{NT}(\mathbf{x}') = 4.32 \quad \%\text{Error} = 0.23\%$
- $C_{\tilde{G}} = 1/[3.16, 5.67] \rightarrow y_c(\mathbf{x}') = 4.42 \quad y_{NT}(\mathbf{x}') = 4.40 \quad \%\text{Error} = 0.45\%$

For these two FOUs, $y_{NT}(\mathbf{x}')$ is very close to $y_c(\mathbf{x}')$. Results are also given in Mendel and Liu (2013) for the discrete cases of $\tilde{F}$ and $\tilde{G}$ for 100, 500, 1000 and 5000 samples of these IT2 FSs. Very similar accuracy results are obtained.

Although this author is not advocating the use of centroid type-reduction + defuzzification, because of the additional computational cost for performing the union of fired-rule output sets, for those who really want to do this kind of type-reduction, y_{NT} gives results that are very close to the ones obtained by performing centroid type-reduction + defuzzification.

Biglarbegian et al. (2008) proposed the (9.155) averaging technique as a replacement for height type-reduction, i.e.,

$$y_{NT}(\mathbf{x}') = \frac{\sum_{l=1}^{M} \bar{y}^l \left[\underline{f}^l(\mathbf{x}') + \bar{f}^l(\mathbf{x}')\right]}{\sum_{l=1}^{M} \left[\underline{f}^l(\mathbf{x}') + \bar{f}^l(\mathbf{x}')\right]} \tag{9.157}$$

Kayacan and Khanesar (2016) used (9.157) to demonstrate noise reduction in an IT2 fuzzy system.

In (9.155)–(9.157), when all IT2 MF uncertainties disappear, they reduce to correct type-1 defuzzification formulas (Exercise 9.15).

9.9.2 Biglarbegian–Melek–Mendel (BMM) Direct Defuzzification

Biglarbegian et al. (2010, 2011) defuzzify directly as follows:

$$y_{\mathrm{BMM}}(\mathbf{x}') = m\frac{\sum_{l=1}^{M} y_l \underline{f}^l(\mathbf{x}')}{\sum_{l=1}^{M} \underline{f}^l(\mathbf{x}')} + n\frac{\sum_{l=1}^{M} y_l \bar{f}^l(\mathbf{x}')}{\sum_{l=1}^{M} \bar{f}^l(\mathbf{x}')} \tag{9.158}$$

Equation (9.158) is a weighted combination of the outputs from two type-1 fuzzy systems, one associated with just the left ends of the firing intervals, and the other associated with just the right ends of the firing intervals. $y_{\mathrm{BMM}}(\mathbf{x}')$ can be interpreted as a linear combination of $y_l^{(0)}(\mathbf{x}')$ and $y_r^{(0)}(\mathbf{x}')$ in (9.142) and (9.145), respectively (see, also Juang and Wang 2015).

In (9.158), when all IT2 MF uncertainties disappear, then it reduces to a correct type-1 defuzzification formula (Exercise 9.16).

Although (9.158) was presented first in the context of an IT2 TSK fuzzy system, where it was called an "*m*–*n* IT2 TSK FLS," there is no reason that it cannot also be applied to an IT2 Mamdani fuzzy system. For an IT2 TSK fuzzy system y_i is the same as $g^i(\mathbf{x})$ in the consequent of an IT2 TSK rule. For an IT2 Mamdani fuzzy system y_i plays the role of the weight in either height or COS defuzzification. In principle, the same y_l does not have to be used in both terms in (9.158).[11]

Khanesar and Mendel (2016) prove that $y_{\mathrm{BMM}}(\mathbf{x}')$ in (9.158), in which $m = n = 1/2$, is a zero-order Maclaurin series approximation to $y_{\mathrm{COS}}(\mathbf{x}')$ in (9.113). They also have a formula for a correction to $y_{\mathrm{BMM}}(\mathbf{x}')$, which when added to $y_{\mathrm{BMM}}(\mathbf{x}')$ provides a first-order Maclaurin series approximation to $y_{\mathrm{COS}}(\mathbf{x}')$.

Formulaic connections between BMM and WM UB are given in Biglarbegian et al. (2010), Mendel et al. (2014, Chap. 6), and between BMM and NT are given in Mendel (2013).

9.10 Summary

So as to see the forest from the trees, Table 9.5 summarizes what has transpired in Sects. 9.6.3, 9.8, 9.9.1 and 9.9.2 in a novel way. Recall, from Example 9.10, that an IT2 fuzzy system can have many more second-order partitions than a type-1 fuzzy system. Within each second-order partition, each IT2 fuzzy system makes use of the endpoints of the firing interval in a different way. Type-reduction + defuzzification uses mixtures of them, WM UBs uses them separately as well as their differences, NT in (9.157) uses their averages, and BMM uses them separately.

[11] Li et al. (2011) have extended (9.158) to the situation where the same y_i does not appear in both terms of (9.158).

Table 9.5 The different ways in which $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are used and combined

Method	How $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are used	Comments
Type-reduction + defuzzification	Different *mixtures* of $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are used depending upon $\mathbf{x}'$	• The mixtures are determined by the switch points L and R obtained from EIASC or EKM algorithms • Examine (9.113) [using (9.111) and (9.112)] to see that $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are combined quadratically in it, including self and cross-product terms • A type-reduced set is obtained • The type-reduced set is defuzzified
WM UBs	All $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are always used, and the *differences* between $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are also used	• Examine (9.149) and (9.150) to see that, when they along with (9.147) and (9.148) are combined, $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ appear quartically in (9.152) • The endpoints of the type-reduced set are bounded from above and below • An approximate type-reduced set is obtained • The approximate type-reduced set is defuzzified
NT	*Averages* of $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are always used	• Examine, e.g., (9.157) to see that $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are combined linearly • The defuzzifed output is obtained directly
BMM	$\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are used *separately* to compute outputs for two type-1 fuzzy systems	• Examine (9.158) to see that $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ are combined quadratically, but only cross-product terms $(\underline{f}^i \times \bar{f}^j)$ appear in it • The defuzzifed output is obtained directly

9.11 Comprehensive Example Continued

This is a continuation of the Sect. 9.7 comprehensive example, so it is advisable for the reader to review that section. Here, the data from that section are used, and $y_{\mathrm{WMUB}}(\mathbf{x}')$, $y_{\mathrm{NT}}(\mathbf{x}')$ and $y_{\mathrm{BMM}}(\mathbf{x}')$ are computed.

Turning first to $y_{\mathrm{WMUB}}(\mathbf{x}')$, beginning with the consequent centroids and firing intervals that are given, respectively, in Table 9.2 for $\mathbf{x}' = \mathrm{col}(-0.3, 0.6)$, $\bar{y}_l(\mathbf{x}')$, $\underline{y}_l(\mathbf{x}')$, $\underline{y}_r(\mathbf{x}')$, $\bar{y}_r(\mathbf{x}')$, $\hat{y}_l(\mathbf{x}')$, $\hat{y}_r(\mathbf{x}')$, and $y_{\mathrm{WMUB}}(\mathbf{x}')$ are computed using (9.142)–(9.150) and (9.152). The results are $y_l^{(0)}(\mathbf{x}') = -0.30$, $y_r^{(M)}(\mathbf{x}') = -0.12$, $y_l^{(M)}(\mathbf{x}') = -0.137$, $y_r^{(0)}(\mathbf{x}) = 0.17$, and:

$$\bar{y}_l(\mathbf{x}') = \min\left\{y_l^{(0)}(\mathbf{x}'),\ y_l^{(M)}(\mathbf{x}')\right\} = \min\{-0.30, -0.137\} = -0.30 \tag{9.159}$$

$$\begin{aligned}
\underline{y}_l(\mathbf{x}') &= \bar{y}_l(\mathbf{x}') - \left[\frac{\sum_{s=1}^4 (\bar{f}^s - \underline{f}^s)}{\sum_{s=1}^4 \bar{f}^s \sum_{s=1}^4 \underline{f}^s} \times \frac{\sum_{s=1}^4 \underline{f}^s (a^s - a^1)\ \sum_{s=1}^4 \bar{f}^s (a^4 - a^s)}{\sum_{s=1}^4 \underline{f}^s (a^s - a^1) + \sum_{s=1}^4 \bar{f}^s (a^4 - a^s)}\right] \\
&= -0.30 - \left[\left(\frac{1.9}{2.175 \times 0.275}\right) \times \left(\frac{0.193 \times 2.255}{0.193 + 2.255}\right)\right] \\
&= -0.30 - \left[\frac{1.9}{0.598} \times \frac{0.435}{2.448}\right] = -0.30 - \frac{0.827}{1.464} = -0.30 - 0.565 \\
&= -0.865
\end{aligned} \tag{9.160}$$

In a similar manner, it follows that:

$$\underline{y}_r(\mathbf{x}') = 0.017 \tag{9.161}$$

$$\bar{y}_r(\mathbf{x}') = 0.865 \tag{9.162}$$

Consequently,

$$\hat{y}_l(\mathbf{x}') = (-0.865 - 0.30)/2 = -0.583 \tag{9.163}$$

$$\hat{y}_r(\mathbf{x}') = (0.017 + 0.865)/2 = 0.441 \tag{9.164}$$

$$y_{\mathrm{WMUB}}(\mathbf{x}') = (-0.583 + 0.441)/2 = -0.071 \approx -0.07 \tag{9.165}$$

Additionally, it follows from (9.153), (9.136) and (9.154), that:

$$\delta(\mathbf{x}') = -0.047 - 0.070 = 0.023 \tag{9.166}$$

$$\delta(\mathbf{x}') = 0.024 \le 0.25[(-0.30 + 0.865) + (0.865 - 0.017)] = 0.353 \tag{9.167}$$

Using (9.163)–(9.165) and (9.134)–(9.136), observe the following:

- It is true that $\underline{y}_l \le y_l^{\text{COS}} \le \bar{y}_l$, i.e., $-0.865 \le -0.717 \le -0.3$
- It is true that $\underline{y}_r \le y_r^{\text{COS}} \le \bar{y}_r$, i.e., $0.017 \le 0.624 \le 0.865$
- It is true that (9.154) is satisfied [see (9.167)]
- $y_{\text{WMUB}}(\mathbf{x}') = -0.07$, and although this is in the ballpark of $y_{\text{COS}}(\mathbf{x}') = -0.047$, it is not a very good approximation to it.

It is once again important to remember that: (1) it is not the closeness of $y_{\text{WMUB}}(\mathbf{x}')$ to $y_{\text{COS}}(\mathbf{x}')$ that is important when the WM UBs are used in an IT2 fuzzy system; (2) $y_{\text{WMUB}}(\mathbf{x}')$ would be used to avoid the iterative computations of type-reduction; and, (3) what is really important is whether or not acceptable system performance is achieved (using the metrics of performance for a specific application) when the WM UBs are used in an IT2 fuzzy system.

Interestingly, there are a lot of computations needed to obtain $y_{\text{WMUB}}(\mathbf{x}')$ by using (9.142)–(9.150), so it is even possible that they take longer to perform than does type reduction.

Turning next to the NT direct defuzzifier in (9.157), two choices are made here for $\bar{y}^i$:

- $\bar{y}^i = g^i(\mathbf{x}')$, where $g^i(\mathbf{x}')$ $(i = 1, \ldots, 4)$ are given in the last column of Table 9.3, so that:

$$\begin{aligned} y_{\text{NT}}(\mathbf{x}') &= \frac{\sum_{i=1}^{4} \bar{y}^i \left[\underline{f}^i(\mathbf{x}') + \bar{f}^i(\mathbf{x}')\right]}{\sum_{i=1}^{4} \left[\underline{f}^i(\mathbf{x}') + \bar{f}^i(\mathbf{x}')\right]} \\ &= \frac{1.75(0+0.405) + 2.75(0.22+0.90) + 3.75(0+0.27) + 4.75(0.055+0.60)}{(0+0.405) + (0.22+0.90) + (0+0.27) + (0.055+0.60)} \\ &= 7.913/2.45 = 3.23 \end{aligned} \tag{9.168}$$

- $\bar{y}^i = (a^i + b^i)/2$, where a^i and b^i $(i = 1, \ldots, 4)$ are given in the last column of Table 9.2, so that $\bar{y}^1 = -0.95$, $\bar{y}^2 = -0.50$, $\bar{y}^3 = 0.50$, $\bar{y}^4 = 0.95$. Consequently,

$$\begin{aligned} y_{\text{NT}}(\mathbf{x}') &= \frac{\sum_{i=1}^{4} \bar{y}^i \left[\underline{f}^i(\mathbf{x}') + \bar{f}^i(\mathbf{x}')\right]}{\sum_{i=1}^{4} \left[\underline{f}^i(\mathbf{x}') + \bar{f}^i(\mathbf{x}')\right]} \\ &= \frac{-0.95(0+0.405) - 0.50(0.22+0.90) + 0.50(0+0.27) + 0.95(0.055+0.60)}{(0+0.405) + (0.22+0.90) + (0+0.27) + (0.055+0.60)} \\ &= -0.188/2.45 = -0.077 \end{aligned} \tag{9.169}$$

Both of these $y_{\mathrm{NT}}(\mathbf{x}')$ values are not close to $y_{\mathrm{COS}}(\mathbf{x}') = -0.047$; however, the admonishment given above about the unimportance of the closeness of $y_{\mathrm{WMUB}}(\mathbf{x}')$ to $y_{\mathrm{COS}}(\mathbf{x}')$ applies here as well. A big difference between $y_{\mathrm{NT}}(\mathbf{x}')$ and $y_{\mathrm{WMUB}}(\mathbf{x}')$ is that the latter are much simpler to compute than the former.

Finally, turning to the BMM direct defuzzifier in (9.158), the same two choices are made for y_i as were just made for $\bar{y}^i$:

- $y_i = g^i(\mathbf{x}')$, where $g^i(\mathbf{x}')$ $(i = 1, \ldots, 4)$ are given in the last column of Table 9.3, so that:

$$y_{\mathrm{BMM}}(\mathbf{x}') = m\frac{\sum_{i=1}^{4} y_i \underline{f}^i(\mathbf{x}')}{\sum_{i=1}^{4} \underline{f}^i(\mathbf{x}')} + n\frac{\sum_{i=1}^{4} y_i \bar{f}^i(\mathbf{x}')}{\sum_{i=1}^{4} \bar{f}^i(\mathbf{x}')} = 3.15m + 3.24n \tag{9.170}$$

- $y_i = (a^i + b^i)/2$, where a^i and b^i $(i = 1, \ldots, 4)$ are given in the last column of Table 9.2, so that $y_1 = -0.95,\ y_2 = -0.50,\ y_3 = 0.50,\ y_4 = 0.95$. Consequently,

$$y_{\mathrm{BMM}}(\mathbf{x}') = m\frac{\sum_{i=1}^{4} y_i \underline{f}^i(\mathbf{x}')}{\sum_{i=1}^{4} \underline{f}^i(\mathbf{x}')} + n\frac{\sum_{i=1}^{4} y_i \bar{f}^i(\mathbf{x}')}{\sum_{i=1}^{4} \bar{f}^i(\mathbf{x}')} = -0.21m - 0.06n \tag{9.171}$$

Observe, in both (9.170) and (9.171), that $y_{\mathrm{BMM}}(\mathbf{x}')$ is not fully specified until m and n are both specified. Because m and n are usually required to be positive numbers, (9.170) has no possibility of being close to $y(\mathbf{x}') = -0.047$; however, (9.171) is easily matched to $y(\mathbf{x}') = -0.047$, e.g., set $n = 0.5$ and $m = 0.081$. Although it has been possible to achieve this exact match at this one value of $\mathbf{x}'$, the admonishment given above about the unimportance of the closeness of $y_{\mathrm{WMUB}}(\mathbf{x}')$ or $y_{\mathrm{NT}}(\mathbf{x}')$ to $y_{\mathrm{COS}}(\mathbf{x}')$ applies here as well. It is also true that the computation of $y_{\mathrm{BMM}}(\mathbf{x}')$ is also much simpler than the computation of $y_{\mathrm{WMUB}}(\mathbf{x}')$.

9.12 IT2 Fuzzy Basis Functions

Just as it was useful to describe the output of a specific type-1 fuzzy system as a fuzzy basis function (FBF) expansion (Sect. 3.8), it is also useful to do the same for the output of a specific IT2 fuzzy system. Because there are many different kinds of IT2 fuzzy systems, this is done in this section for four of them that are distinctly different, namely: IT2 Mamdani fuzzy system with COS type-reduction + defuzzification, A2-C0 unnormalized IT2 TSK fuzzy system, NT direct defuzzification fuzzy system and BMM direct defuzzification fuzzy system. Exercises 9.18–9.23 ask the readers to obtain the IT2 FBFs for other kinds of IT2 fuzzy systems. Because there can be three kinds of fuzzifiers for each of these fuzzy systems, it is best to express the IT2 FBSs in terms of the firing intervals. One can then use (9.25) and (9.26), or (9.51) and (9.52), or (9.82) and (9.83) to express the firing intervals in terms of lower and upper MFs.

Example 9.13 **(IT2 Mamdani fuzzy system with COS type-reduction + defuzzification)**: There are different ways to express $y_{\text{COS}}(\mathbf{x}')$ in (9.113) because there are different ways to express $y_l^{\text{COS}}(\mathbf{x}')$ and $y_r^{\text{COS}}(\mathbf{x}')$. A direct approach is to use (9.111) and (9.112) in (9.113); however, it is the opinion of this author that the resulting expression is more complicated than it has to be. Instead, (8.18) and (8.20) are used in (9.113) [in which $c_i \equiv \underline{f}^i(\mathbf{x})$ and $d_i \equiv \bar{f}^i(\mathbf{x})$] to obtain the following very compact *IT2 FBF expansion* for[12] $y_{\text{COS}}(\mathbf{x})$:

$$y_{\text{COS}}(\mathbf{x}) = \frac{1}{2} y_l^{\text{COS}}(\mathbf{x}) + \frac{1}{2} y_r^{\text{COS}}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{M} c_l(\tilde{G}^i)\,\phi_l^i(\mathbf{x}) + \frac{1}{2}\sum_{i=1}^{M} c_r(\tilde{G}^i)\,\phi_r^i(\mathbf{x}) \tag{9.172}$$

In (9.172), the *IT2 fuzzy basis functions* (IT2 FBFs) are $(i = 1, \ldots, M)$

$$\phi_l^i(\mathbf{x}) = \frac{\delta_l^i \bar{f}^i(\mathbf{x}) + (1 - \delta_l^i)\underline{f}^i(\mathbf{x})}{\sum_{i=1}^{M} [\delta_l^i \bar{f}^i(\mathbf{x}) + (1 - \delta_l^i)\underline{f}^i(\mathbf{x})]} \tag{9.173}$$

$$\phi_r^i(\mathbf{x}) = \frac{\delta_r^i \bar{f}^i(\mathbf{x}) + (1 - \delta_r^i)\underline{f}^i(\mathbf{x})}{\sum_{i=1}^{M} [\delta_r^i \bar{f}^i(\mathbf{x}) + (1 - \delta_r^i)\underline{f}^i(\mathbf{x})} \tag{9.174}$$

In (9.173)

$$\delta_l^i = \begin{cases} 1 & c_l(\tilde{G}^i) \le y_l^{\text{COS}}(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{9.175}$$

and in (9.174)

$$\delta_r^i = \begin{cases} 1 & c_r(\tilde{G}^i) \ge y_r^{\text{COS}}(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{9.176}$$

Observe that there are $2M$ IT2 FBFs, and both $\phi_l^i(\mathbf{x})$ and $\phi_r^i(\mathbf{x})$ depend on a mixture of the lower and upper firing intervals.

Although it has been possible to write formulas for $y_l^{\text{COS}}(\mathbf{x})$ and $y_r^{\text{COS}}(\mathbf{x})$, these quantities cannot actually be computed using (9.173)–(9.176), because $y_l^{\text{COS}}(\mathbf{x})$ and $y_r^{\text{COS}}(\mathbf{x})$ are not known ahead of time.[13] These latter quantities can only be determined by first using iterative EIASC or EKM algorithms to determine the switch

[12]The IT2 FBFs are shown as a function of $\mathbf{x}$ rather than of $\mathbf{x}'$ since they are valid for $\mathbf{x} \in \mathbf{X}$.

[13]An equivalent way of saying this is that in (9.111) and (9.112) L and R are not known ahead of time.

points, followed by their use in (9.111) and (9.112). This is very different from the T1 FBF expansion formulas that are in Sect. 3.8 which not only *describe* the T1 FBF expansions, but can also be used to *compute* them because everything is known in them. Note, however, that once $y_l^{\mathrm{COS}}(\mathbf{x})$ and $y_r^{\mathrm{COS}}(\mathbf{x})$ are known, then it still may be very useful to organize and describe $y_{\mathrm{COS}}(\mathbf{x})$ using (9.172)–(9.176). These formulas will, for example, be used in the derivative-based design method that is described in Sect. 10.2.3.

Importantly, when $y_{\mathrm{COS}}(\mathbf{x})$ is expressed using (9.172)–(9.176) rule-reordering is not a concern, because both of the summations in (9.172) range from 1 to M (i.e., over all of the rules). On the other hand, if the representations for $y_l^{\mathrm{COS}}(\mathbf{x})$ and $y_r^{\mathrm{COS}}(\mathbf{x})$ that are in (9.111) and (9.112) are used to obtain IT2 FBF formulas, then one must be very careful, because those equations are not in a rule-ordered format, due to needing to use EIASC or EKM algorithms to compute L and R. These two equations have to be re-expressed in a rule-order before, e.g., derivatives of them can be taken with respect to a specific FOU parameter, because it is only then that one knows exactly where a FOU parameter appears in those equations.

Example 9.14 **(Unnormalized A2-C0 IT2 TSK fuzzy system)**: Substituting (9.115)–(9.117) into (9.118), one finds:

$$y_{\mathrm{TSK}}^{U}(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{M}\left[c_0^l + \sum_{j=1}^{p} c_j^l x_j\right]\underline{f}^l(\mathbf{x}) + \frac{1}{2}\sum_{l=1}^{M}\left[c_0^l + \sum_{j=1}^{p} c_j^l x_j\right]\bar{f}^l(\mathbf{x})] \tag{9.177}$$

Equation (9.177) can be expressed as:

$$y_{\mathrm{TSK}}^{U}(\mathbf{x}) = \sum_{l=1}^{M}\sum_{j=0}^{p} c_j^l \phi_j^l(\mathbf{x}) \tag{9.178}$$

where the IT2 FBFs are $(l = 1,\ldots,M;\ j = 0,1,\ldots,p;\ x_0 \equiv 1)$:

$$\phi_j^l(\mathbf{x}) = \left[\frac{\underline{f}^l(\mathbf{x}) + \bar{f}^l(\mathbf{x})}{2}\right] \times x_j \tag{9.179}$$

Observe that there are $M(p+1)$ IT2 FBFs all of which depend on the average of the lower and upper firing intervals.

Example 9.15 **(NT direct defuzzification fuzzy system)**: (9.157) can be expressed as:

$$y_{\mathrm{NT}}(\mathbf{x}) = \sum_{l=1}^{M} \bar{y}^l \phi_l(\mathbf{x}) \tag{9.180}$$

where the IT2 FBFs are $(l = 1, \ldots, M)$:

$$\phi_l(\mathbf{x}) = \frac{\underline{f}^l(\mathbf{x}) + \bar{f}^l(\mathbf{x})}{\sum_{l=1}^{M}\left[\underline{f}^l(\mathbf{x}) + \bar{f}^l(\mathbf{x})\right]} \tag{9.181}$$

Observe that there are M IT2 FBFs all of which depend on a normalized average of the lower and upper firing intervals.

Example 9.16 **(BMM direct defuzzification fuzzy system)**: (9.158) is repeated here for the convenience of the reader:

$$y_{\mathrm{BMM}}(\mathbf{x}) = m\frac{\sum_{i=1}^{M} y_i \underline{f}^i(\mathbf{x})}{\sum_{i=1}^{M} \underline{f}^i(\mathbf{x})} + n\frac{\sum_{i=1}^{M} y_i \bar{f}^i(\mathbf{x})}{\sum_{i=1}^{M} \bar{f}^i(\mathbf{x})} \tag{9.182}$$

Equation (9.182) can be expressed as:

$$y_{\mathrm{BMM}}(\mathbf{x}) = \sum_{i=1}^{M} y_i\, \phi_l^i(\mathbf{x}) + \sum_{i=1}^{M} y_i\, \phi_r^i(\mathbf{x}) \tag{9.183}$$

where the IT2 FBFs are $(i = 1, \ldots, M)$:

$$\phi_l^i(\mathbf{x}) = \frac{m\underline{f}^i(\mathbf{x})}{\sum_{i=1}^{M} \underline{f}^i(\mathbf{x})} \tag{9.184}$$

$$\phi_r^i(\mathbf{x}) = \frac{n\bar{f}^i(\mathbf{x})}{\sum_{i=1}^{M} \bar{f}^i(\mathbf{x})} \tag{9.185}$$

Observe that there are 2M IT2 FBFs the first M of which depend only on the LMFs and the next M of which depend only on the UMFs. $y_{\mathrm{BMM}}(\mathbf{x}')$ decouples the dependence of its FBFs on both the lower and upper MFs.

Examples 9.13–9.16 demonstrate that IT2 FBFs make use of firing intervals in different ways (as was already noted in Table 9.5), and so this is what really distinguishes one kind of IT2 fuzzy system from another.

9.13 Remarks and Insights

As was the case for type-1 fuzzy systems, before getting into the designs of IT2 fuzzy systems as well as some applications for them, there are many aspects of these systems that deserve some discussions. In this section, such discussions are provided about layered architecture interpretations of an IT2 fuzzy system, fundamental differences between type-1 and IT2 fuzzy systems, universal approximation by IT2 fuzzy systems, continuity of IT2 fuzzy systems, rule explosion and some ways to control it, rule interpretability, and historical notes.

9.13.1 Layered Architecture Interpretations of an IT2 Fuzzy System

As was true for a type-1 fuzzy system, it should be very clear that there is a flow to the computations in all IT2 fuzzy systems, as is evidenced by the heavy-arrowed lines in Fig. 9.1a, b. For example, for an IT2 Mamdani fuzzy system with non-singleton fuzzification ($i = 1, \ldots, p$; $l = 1, .., M$):

- Inputs are fuzzified: $x_i' \rightarrow \mu_{\tilde{X}_i}(x_i|x_i')$
- Firing intervals are computed for each rule by the inference engine:

$$\left(\mu_{\tilde{X}_i}(x_i|x_i'),\ \underline{\mu}_{\tilde{F}_i^l}(x_i),\ \bar{\mu}_{\tilde{F}_i^l}(x_i)\right) \rightarrow [\underline{f}_i^l(x_i'),\ \bar{f}_i^l(x_i')] \rightarrow [\underline{f}^l(\mathbf{x}'),\ \bar{f}^l(\mathbf{x}')]$$

- Fired-rule outputs may or may or not be combined (not shown on Fig. 9.1a, b); if they are combined, then $(y \in Y)$:

$$\left([\underline{f}^l(\mathbf{x}'),\ \bar{f}^l(\mathbf{x}')],\ [\underline{\mu}_{\tilde{G}^l}(y),\ \bar{\mu}_{\tilde{G}^l}(y)]\right) \rightarrow [\underline{\mu}_{\tilde{B}^l}(y|\mathbf{x}'),\ \bar{\mu}_{\tilde{B}^l}(y|\mathbf{x}')] \rightarrow [\underline{\mu}_{\tilde{B}}(y|\mathbf{x}'),\ \bar{\mu}_{\tilde{B}}(y|\mathbf{x}')]$$

- Combined fired-rule outputs sets are type reduced, using centroid type-reduction, into a type-1 interval fuzzy number:

$$[\underline{\mu}_{\tilde{B}}(y|\mathbf{x}'),\ \bar{\mu}_{\tilde{B}}(y|\mathbf{x}')] \rightarrow C_{\tilde{B}} = 1/[c_l(\tilde{B}),\ c_r(\tilde{B})];$$

  or, if fired-rule output sets are not combined, then firing intervals and, e.g., centroids of their respective consequent IT2 FSs, are type-reduced (using, e.g., COS type reduction) into a type-1 interval fuzzy number:

$$\left([\underline{f}^l(\mathbf{x}'),\ \bar{f}^l(\mathbf{x}')],\ [c_l(\tilde{G}^i),\ c_r(\tilde{G}^i)]\right) \rightarrow Y_{\mathrm{COS}}(\mathbf{x}') = 1/[y_l^{\mathrm{COS}}(\mathbf{x}'),\ y_r^{\mathrm{COS}}(\mathbf{x}')]$$

- The type-reduced set is defuzzified into a crisp number:

$$[c_l(\tilde{B}),\ c_r(\tilde{B})]] \rightarrow y_c(\mathbf{x}') \text{ or } [y_l^{\mathrm{COS}}(\mathbf{x}'),\ y_r^{\mathrm{COS}}(\mathbf{x}')] \rightarrow y_{\mathrm{COS}}(\mathbf{x}')$$

This flow of computations can also be interpreted as a flow through a layered architecture or network (e.g., Juang and Tsao 2008, Fig. 1; Castro et al. 2009, Fig. 7; Aliev et al. 2011, Fig. 4; Lin and Chen 2011, Fig. 3; Mendel et al. 2014, Fig. 3.9).

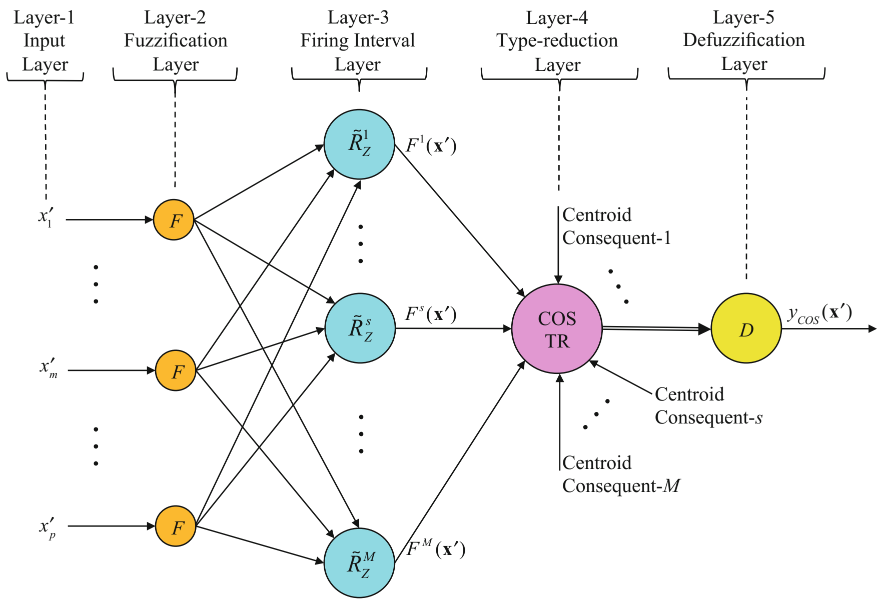


Fig. 9.16 Layered architecture summary of IT2 Mamdani fuzzy system computations that use COS type-reduction (Mendel et al. (2014, p. 99, © 2014 Wiley)

Figure 9.16 is an example of one of these figures. Layer 2 is where the inputs are fuzzified (singleton, T1 non-singleton or IT2 non-singleton). Layer 3 is where the firing interval is computed for each of the M rules. Layer 4 is where type-reduction is performed and is illustrated for COS type-reduction, for which rule consequent centroids, which have been stored in memory, also have to be used. Layer 5 is where defuzzification is performed. Note that if centroid type-reduction is used two additional layers have to be inserted between layers 3 and 4. In the first of these, fired-rule output sets are computed for each of the M rules, and in the second of these the fired-rule output sets are aggregated by means of the union operation.

Similar figures can be constructed for the other kinds of IT2 fuzzy systems (Exercise 9.24).

When the antecedents of an IT2 fuzzy system include some time-delayed versions of the consequent, then such an IT2 fuzzy system is called *recurrent*. This kind of a IT2 fuzzy system (Juang et al. 2009) may give better results than the more usual (static) IT2 fuzzy system and recurrent T1 fuzzy systems for identification of nonlinear dynamical systems and forecasting of time series.

Although some authors give the Fig. 9.16 layered architecture a “neural” designation, there is nothing neural about it. Just as a T1 fuzzy system is not a neural network, an IT2 fuzzy system is not a neural network.

9.13.2 Fundamental Differences Between T1 and IT2 Fuzzy Systems

Wu (2011, 2012, 2013) was the first to carefully explain two fundamental differences between T1 and IT2 fuzzy systems that use type-reduction (for a Mamdani fuzzy system with COS or height type-reduction + defuzzification and a normalized TSK fuzzy system):

1. *Adaptiveness*, meaning that *the endpoints of the firing interval used to compute the endpoints of the type-reduced set change as inputs change*. Although this has not been illustrated in our comprehensive example, because it was only for one input, it has already been explained in Example 9.10 in terms of second-order partitions. If the reader works out Exercise 9.13, which requests the computations in Sect. 9.7 to be repeated for a different input, then they will also observe adaptiveness explicitly.
2. *Novelty*, meaning that *different endpoints of a firing interval may be used simultaneously in computing each endpoint of the type-reduced set*. This can be seen in (9.134) and (9.135) where $\bar{f}^1$ and $\underline{f}^4$ are used to compute $y_l^{\mathrm{COS}}(\mathbf{x}')$, but $\underline{f}^1$ and $\bar{f}^4$ are used to compute $y_r^{\mathrm{COS}}(\mathbf{x}')$. Definition 9.10 has formalized novelty in terms of *IT2 novelty rule partitions*.

As a result of these two properties, an IT2 fuzzy system that uses type-reduction can implement a more complex control surface that cannot be achieved by a T1 fuzzy system. This may also be true for IT2 fuzzy systems that do not use type-reduction (see Fig. 9.17), "may" because *adaptiveness holds for all IT2 fuzzy systems, but novelty only holds for those that use type-reduction.*

9.13.3 Universal Approximation by IT2 Fuzzy Systems

Recall (Sect. 3.9.2) that universal approximation for a fuzzy system means it can uniformly approximate any real continuous function on a compact domain to arbitrary degree of accuracy. Unlike the very large literature about how well a T1 fuzzy system approximates an unknown function (see Sect. 3.9.2), there is a rather small literature about how well an IT2 fuzzy system approximates an unknown function. Perhaps this is due to the many different kinds of IT2 fuzzy systems that are now available (some with type-reduction and others without type-reduction), or the difficulty in examining universal approximation for IT2 fuzzy systems, or the existence nature of universal approximation theorems which do not provide help in the actual designs of IT2 fuzzy systems.

Ying (2008, 2009) uses the Weierstrass theorem from real analysis to demonstrate universal approximation for both IT2 Mamdani and TSK fuzzy systems, the latter only for rule consequents that are linear combinations of their input variables; and, Castillo et al. (2013) use the Stone-Weierstrass theorem from real analysis to demonstrate universal approximation for the A2-C0 and A2-C1 IT2 TSK fuzzy systems.

9.13.4 Continuity of IT2 Fuzzy Systems

Wu and Mendel (2011) examine the continuity of the output of different kinds of IT2 fuzzy systems. This is very technical and so only their findings[14] will be stated here. Their findings are:

1. An IT2 Mamdani fuzzy system that uses COS type-reduction + defuzzification:
   (a) Is *continuous* (see footnote 19 in Chap. 3) as long as its input domain is fully covered by its antecedent's lower *and* upper MFs.
   (b) Has a *jump discontinuity*[15] if the input domain is fully covered by the UMFs, there exists at least one input variable value that is not covered by the LMFs, and all rules have different consequents.
   (c) Has a *gap discontinuity* (see footnotes 18 and 20 in Chap. 3) if and only if there exists at least one input variable value that is not covered by the UMFs.
2. An IT2 fuzzy system that uses WM UB approximate COS type-reduction and defuzzification:
   (a) Is *continuous* as long as its input domain is fully covered by its antecedent's LMFs.
   (b) Cannot have *jump discontinuities.*
   (c) Has a *gap discontinuity* if and only if there exists at least one input variable value that is not covered by the LMFs.
3. An IT2 fuzzy system that uses NT direct defuzzification:
   (a) Is *continuous* as long as its input domain is fully covered by its antecedent's UMFs.
   (b) Cannot have *jump discontinuities.*
   (c) Has a *gap discontinuity* if and only if there exists at least one input variable value that is not covered by the UMFs.

[14]Proofs of these findings are provided in Wu and Mendel (2011).

[15]A function $f(x)$ has a *jump discontinuity* at c if: $f(c)$ is defined but $\lim_{x \to c^+} f(x) \neq \lim_{x \to c^-} f(x)$, i.e. both $f(c)$ and $f(c+\delta)$ are defined, but $f(c+\delta)$ does not approach $f(c)$ as δ approaches 0.

4. An IT2 fuzzy system that uses BMM direct defuzzification:
   (a) Is *continuous* as long as its input domain is fully covered by its antecedent's LMFs.
   (b) Cannot have *jump discontinuities.*
   (c) Has a *gap discontinuity* if and only if there exists at least one input variable value that is not covered by the LMFs.

Observe that it is only an IT2 fuzzy system that uses COS type-reduction + defuzzification that has full coverage conditions on both the lower and upper MFs. For the other three kinds of IT2 fuzzy systems, if full coverage occurs for just the LMFs (UMFs) then continuity occurs, or, if full coverage does not occur for just the LMFs (UMFs) then a gap discontinuity occurs. Since these are the only two situations that are possible for either the LMFs or UMFs, a jump discontinuity cannot occur for these three other IT2 fuzzy systems.

Example 9.17 (Adapted from Wu and Mendel 2011, pp. 183–185) This example illustrates the input–output mappings for four IT2 fuzzy systems with two inputs. Figure 9.17a depicts three FOUs for each domain and for four situations. Table 9.6 provides the nine rules $\tilde{R}_Z^l$ $(l = 1, \ldots, 9)$ that were used for these fuzzy systems. Each entry in the table corresponds to the support of the centroid of a rule's consequent, $[c_l(\tilde{G}^i),\ c_r(\tilde{G}^i)]$.

Examining[16] Fig. 9.17b, which is for *COS type-reduction + defuzzification*, observe that:

- When both the input UMFs and LMFs fully cover the input domain, as shown in the first column of Fig. 9.17b, then the corresponding input–output mapping is continuous, in agreement with Item 1a above.
- When the input domain is fully covered by the UMFs but at least one point in the input domain is not covered by the LMFs, as shown in the middle two columns of Fig. 9.17a, the corresponding input–output mapping has jump discontinuities [e.g., in the second column of Fig. 9.17a, when $x_2 \in [-0.7, -0.4] \cup [0.4, 0.7]$], in agreement with item 1b. A blowup of the second figure in Fig. 9.17b is depicted in Fig. 9.18. A detailed mathematical explanation for why this happens when $x_1 = 0.1$ and $x_2 = 0.4$ is given in Wu and Mendel (2011, p. 185) where it is shown that $y_{\text{COS}}(0.1, 0.4) = 5.5$ and $y_{\text{COS}}(0.1 + \delta, 0.4) = 7$ (Exercise 9.25). Observe also, from Figs. 9.17a and 9.18, that even though it is the x_2 domain that is not fully covered by the LMFs, the jump discontinuities happen in the domain of x_1.
- When the input UMFs do not fully cover the input domain, as shown in the last column of Fig. 9.17a, the corresponding input–output mapping has gap discontinuities, in agreement with item 1c.

[16]The colorized Figs. 9.17 and 9.18 were provided by Dongrui Wu.

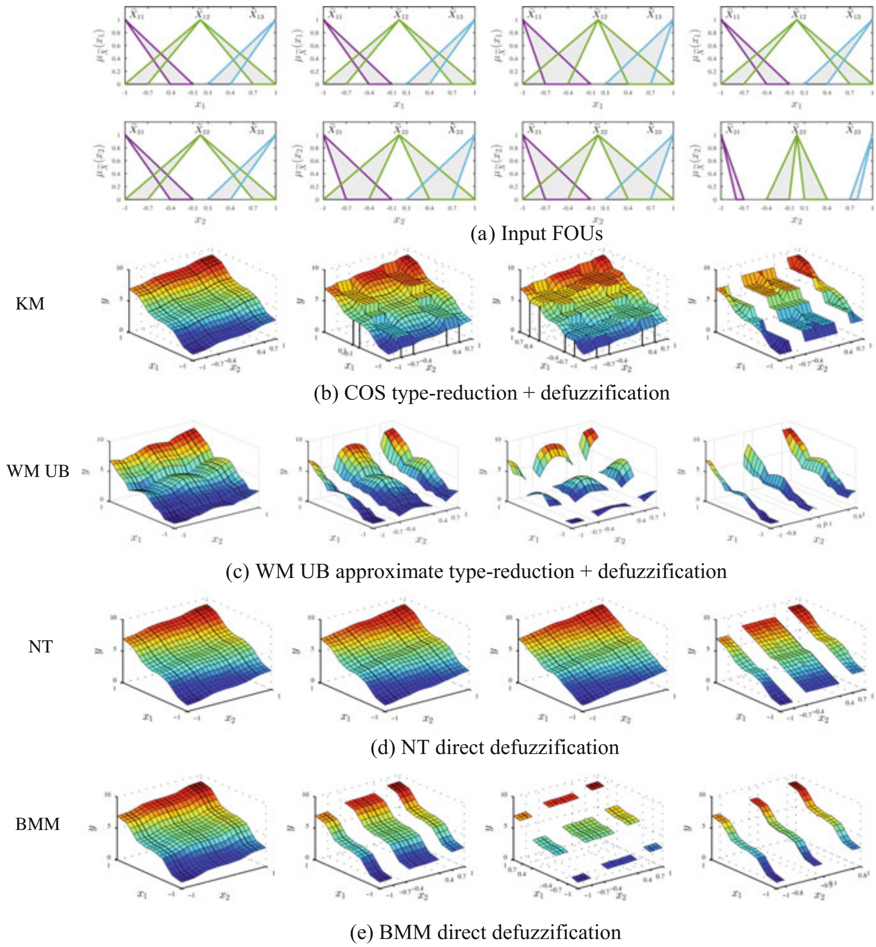


Fig. 9.17 Example input–output mappings of IT2 fuzzy systems with two inputs. **a** Input FOUs for four coverage situations, **b** COS type-reduction + defuzzification, **c** WM UB approximate type-reduction + defuzzification, **d** NT direct defuzzification and **e** BMM direct defuzzification (Wu and Mendel 2011, © 2011 IEEE)

Table 9.6 Rule-base for the IT2 fuzzy systems shown in Fig. 9.17 (Wu and Mendel 2011); the numbers are the supports of the centroids of the rule consequents

x_1/x_2	$\tilde{X}_{21}$	$\tilde{X}_{22}$	$\tilde{X}_{23}$
$\tilde{X}_{11}$	$\tilde{R}_Z^1 : [0.8, 1.2]$	$\tilde{R}_Z^2 : [1.8, 2.2]$	$\tilde{R}_Z^3 : [2.8, 3.2]$
$\tilde{X}_{12}$	$\tilde{R}_Z^4 : [3.8, 4.2]$	$\tilde{R}_Z^5 : [4.8, 5.2]$	$\tilde{R}_Z^6 : [5.8, 6.2]$
$\tilde{X}_{13}$	$\tilde{R}_Z^7 : [6.8, 7.2]$	$\tilde{R}_Z^8 : [7.8, 8.2]$	$\tilde{R}_Z^9 : [8.8, 9.2]$

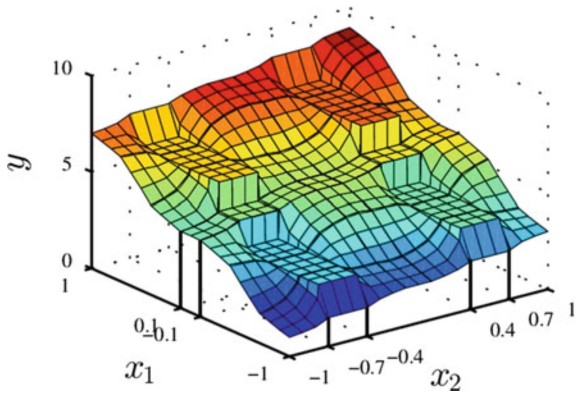


Fig. 9.18 Detailed illustration of jump discontinuities. The input FOUs are shown in the second column of Fig. 9.17a (Wu and Mendel 2011, © 2011 IEEE)

Examining Fig. 9.17c, which is for *WM UB approximate type-reduction + defuzzification*, observe that:

- Jump discontinuities do not occur even when a domain is not fully covered by LMFs., in agreement with Item 2b.
- Gap discontinuities occur more easily than in COS type-reduction + defuzzification, because full coverage of a domain by LMFs is less likely to occur than is full coverage of a domain by UMFs.

Examining Fig. 9.17d, which is for *NT direct defuzzification*, observe that:

- Jump discontinuities do not occur even when a domain is not fully covered by UMFs, in agreement with Item 3b.
- Gap discontinuities occur in the same way as they do for COS type-reduction + defuzzification, and input–output mappings are generally less complex than those obtained from COS type-reduction + defuzzification.

Examining Fig. 9.17e, which is for *BMM direct defuzzification*, observe that:

- Jump discontinuities do not occur even when a domain is not fully covered by UMFs, in agreement with Item 4b.
- Gap discontinuities occur more easily than in COS type-reduction + defuzzification, and the gaps in the input–output mappings are larger than those from COS type-reduction + defuzzification.

The fact that continuity for three of the four IT2 fuzzy systems (NT excluded) requires full coverage of each input's domain by the LMFs was a surprise. It means that when FOU parameters are optimized (as explained in Sect. 10.2), and it is required that the output mapping of one of these IT2 fuzzy systems must be continuous, then the FOU parameters must be constrained so its LMFs and UMFs do not have gaps. When triangle or trapezoidal LMFs and UMFs are used, there can be continuity problems, and constraints have to be imposed on these MFs to avoid them. Interestingly, *when Gaussian LMFs and UMFs are used, coverage gaps can never occur and so continuity is not as issue*. This is arguably a pretty strong reason to use Gaussian LMFs and UMFs.

9.13.5 Rule Explosion and Some Ways to Control It

Recall that "rule explosion" refers to the geometric increase in the number of rules that can occur either as the number of antecedents increase or as the number of MFs for each antecedent increase. Section 3.9.4 mentions that a Singular Value Decomposition (SVD)–QR method can be used to extract the most important type-1 fuzzy rules from a given rule base. That method, which is explained in Sect. 4.2.4, has also been extended to IT2 fuzzy rules (Liang and Mendel 2000b) (see, also Zhou et al. 2009), and it will be explained in Sect. 10.2.4.

Section 3.9.4 also lists five other approaches for controlling rule explosion. Next, this list is examined for its applicability to IT2 fuzzy systems:

1. Fix the number of rules ahead of time: this can also be done for an IT2 fuzzy system.
2. Use rule interpolation when data produce a sparse set of rules, i.e., when rules do not cover all of $X_1 \times X_2 \times \cdots \times X_p$: to-date this has not been tried for IT2 fuzzy systems.
3. Map the IT2 Zadeh rules in (9.1), that could also be called the "and-configuration" of rules, into a "union configuration" that is theoretically equivalent to the and-configuration when the equivalence analysis is performed using crisp set theory: to-date this has not been tried for IT2 fuzzy systems.
4. Combine input variables using sensor fusion: this can also be done for an IT2 fuzzy system (e.g., Jamshidi 1997, Sect. 8.3.2; Kumar et al. 2012).
5. Use a hierarchical structure for the fuzzy system (see Example 3.18 and Fig. 3.19: this can also be done for an IT2 fuzzy system (e.g., Liu et al. 2012). Hagras (2004) takes a different approach to controlling rule explosion in an IT2 hierarchical FLC system and, he:

> "… hierarchically decomposes the control problem by breaking down the input space for analysis by sharing it amongst low-level type-2 behaviors. Each behavior responds to specific types of situations, and then integrates the recommendations of these behaviors via a higher level type-2 coordination layer. Each behavior is an independent and self-contained IT2 FLC with a small number of inputs and outputs and a small rule base and it serves a single purpose (e.g., edge-following or obstacle-avoidance) while operating in a reactive fashion. … The hierarchical fuzzy systems have a nice property that the total number of rules[17] increases linearly rather than exponentially as in the single rule-based FLC."

Of course, this approach can only be applied to situations that lend themselves to this kind of decomposition.

Section 5.2 has the following paragraph that is pertinent to rule reduction:

> Linguistic uncertainty appears to be useful in that it lets the 0–10 range be covered with a much smaller number of labels than without it. Put another way, in the context of firing rules in a fuzzy system, *uncertainty can fire rules*. This cannot occur in the framework of a

[17]This kind of hierarchical approach can also be used in a T1 fuzzy system.

> type-1 fuzzy system; but it can occur in the framework of a type-2 fuzzy system. Additionally, *uncertainty can be used to control the rule explosion* that is so common in a fuzzy system. If, for example, one ignored uncertainty, and had rules with two antecedents, each of which has six labels, it would take 36 rules to completely describe the fuzzy rule base. On the other hand, using three labels for each antecedent requires a rule base with only nine rules. This is a 75% reduction in the size of the rule base. For rules with more than two antecedents the rule reduction is even greater.

So, the broader coverage of an input's domain by FOUs has the potential to be a way to control rule explosion, i.e., it may be that a variable does not have to be partitioned as fine as it might have to be for a T1 fuzzy system when FOUs are used in an IT2 fuzzy system.

9.13.6 Rule Interpretability

Everything that is in Sect. 3.9.5 about rule interpretability for T1 fuzzy systems is also applicable for rule interpretability for IT2 fuzzy systems, with one notable caveat: for *semantics-based interpretability*, if rule-words are modeled using IT2 FSs, then it is this author's belief that those models should be based on data that are collected from a group of subjects in order for the model to capture the inter-and intra-uncertainties of a word (see Sect. 5.2). Even when this is done, if rules have too many antecedents and there are too many rules, it is arguably very questionable as to the interpretability of IT2 rules. It may be, as has just been mentioned, that by using FOUs to cover the domain of a variable with a small number of words, fewer rules are needed and so interpretability is more achievable when IT2 fuzzy rules are used then when T1 fuzzy rules are used.

9.13.7 Historical Notes

Gorzalczany published some pioneering works on fuzzy systems that use IVFSs. His papers have results that are comparable to some of the ones that are in this chapter. More specifically, in Gorzalczany (1987):

1. He has figures that show shaded regions, and although he did not call them "footprints of uncertainty" (he provided no names for them) that's what they are.
2. He provides no detailed derivations of results, and instead gives them as "definitions," although he states (p. 3): that they may be derived in a formal way by applying the Extension Principle. Gorzalczany (1983), which is his Ph.D. Thesis (it is in Polish), is also referred to as a source for the details.
3. His "compatibility measure" between an input fuzzy set and its corresponding antecedent fuzzy set is analogous to our firing interval; however, the lower and upper values for his interval are normalized, whereas ours are not.

4. His "compatibility measure" allows for singleton and non-singleton fuzzifications, and so he may have been among the first researchers, or even the first researcher, to examine non-singleton fuzzification.
5. He states that a "verbal label" may be assigned to input fuzzy sets [e.g., (p. 10) "x is slightly bigger than zero"], and so he is computing with words, long before (Zadeh 1996) coined this phrase.
6. He always combines fired-rule output sets by using the union operation, leading to one aggregated IVFS for a set of rules.
7. He does not indicate how one goes from this aggregated IVFS to a number, arguably because (as is stated in his abstract) he is interested in verbal decision algorithms; however, no method is given for going from the aggregated IVFS to a verbal decision.

   In Dziech and Gorzalczany (1987):
8. As explained in Mendel (2007, p. 100)][18], Dziech and Gorzalczany introduced the following interesting function of[19] $\underline{\mu}_{\tilde{B}}(y)$ and $\bar{\mu}_{\tilde{B}}(y)$ $(y \in Y)$:

$$f(y) \equiv \frac{1}{2}\left[\underline{\mu}_{\tilde{B}}(y) + \bar{\mu}_{\tilde{B}}(y)\right] \times \left\{1 - \left[\bar{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)\right]\right\} \tag{9.186}$$

   in which $[\bar{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]$ is called the "bandwidth" of $\mu_{\tilde{B}}(y)$. Two ways to compute $y(\mathbf{x}')$ from the type-1 FS $f(y)$ are then suggested by them, namely:

$$y_1(\mathbf{x}') = \underset{y \in Y}{\operatorname{argmax}} f(y) \tag{9.187}$$

$$y_2(\mathbf{x}') = \operatorname{median}(f(y)) \tag{9.188}$$

   Observe that when all sources of uncertainty disappear, so that $\underline{\mu}_{\tilde{B}}(y) = \bar{\mu}_{\tilde{B}}(y) = \mu_B(y)$, then $f(y) = \mu_B(y)$ as required by the design requirement, that when all MF uncertainties disappear, an IT2 fuzzy system must reduce to a T1 fuzzy system.
9. Note that, because $f(y)$ is a T1 FS it qualifies to be called a type-reduced set, albeit a very different kind of type-reduced set from the centroid type-reduced set. It remains to be established if there is a connection between these two kinds of type-reduced sets.

Niewiadomski et al. (2006) have defined four other kinds of type-reduction (TR) methods: $\mathrm{TR}_{\mathrm{opt}}(\tilde{B})$, $\mathrm{TR}_{\mathrm{pes}}(\tilde{B})$, $\mathrm{TR}_{\mathrm{re}}(\tilde{B})$ and $\mathrm{TR}_{\mathrm{rew}}(\tilde{B})$, where the lower indices mean: opt = optimistic, pes = pessimistic, re = realistic, and rew = realistic-weighted, and $(y \in Y)$:

[18]This reference incorrectly shows Gorzalczany as the sole author of this paper.

[19]Their function is explained here using this book's notation, which is quite different from their notation.

$$\mathrm{TR}_{\mathrm{opt}}(\tilde{B}) = \bar{\mu}_{\tilde{B}}(y) \tag{9.189}$$

$$\mathrm{TR}_{\mathrm{pes}}(\tilde{B}) = \underline{\mu}_{\tilde{B}}(y) \tag{9.190}$$

$$\mathrm{TR}_{\mathrm{re}}(\tilde{B}) = \frac{1}{2}\left[\underline{\mu}_{\tilde{B}}(y) + \bar{\mu}_{\tilde{B}}(y)\right] \tag{9.191}$$

$$\mathrm{TR}_{\mathrm{rew}}(\tilde{B}) = w\underline{\mu}_{\tilde{B}}(y) + (1-w)\bar{\mu}_{\tilde{B}}(y) \tag{9.192}$$

Each of the functions in (9.189)–(9.192) is a T1 FS (and so each qualifies to be called a type-reduced set). Observe that when all sources of uncertainty disappear, so that $\underline{\mu}_{\tilde{B}}(y) = \bar{\mu}_{\tilde{B}}(y) = \mu_B(y)$, then

$$\mathrm{TR}_{\mathrm{opt}}(\tilde{B}) = \mathrm{TR}_{\mathrm{pes}}(\tilde{B}) = \mathrm{TR}_{\mathrm{re}}(\tilde{B}) = \mathrm{TR}_{\mathrm{rew}}(\tilde{B}) = \mu_B(y) \quad \forall y \in Y \tag{9.193}$$

as required by the design requirement stated above.

All four of these type-reduction methods require that fired rule IT2 FSs must first be unioned so as to obtain the aggregated IT2 output set $\tilde{B}$, and so they are not applicable replacements for height or COS type-reduction, which do not require such an aggregation. $\mathrm{TR}_{\mathrm{opt}}(\tilde{B})$ ignores all LMFs and $\mathrm{TR}_{\mathrm{pes}}(\tilde{B})$ ignores all UMFs, both of which arguably do not seem like good ideas, since MF uncertainties are contained in the FOU, and, if the MF uncertainties of an IT2 fuzzy system are to be reflected in the crisp output, that output needs to be made a function of both the LMFs and UMFs.

$\mathrm{TR}_{\mathrm{re}}(\tilde{B})$ in (9.191) and $y_{\mathrm{NT}}(\mathbf{x}')$ in (9.155) are directly related, as: $y_{\mathrm{NT}}(\mathbf{x}') = \mathrm{COG}(\mathrm{TR}_{\mathrm{re}}(\tilde{B}))$. Although (9.192) seems similar to BMM, they are not related, because BMM is a direct defuzzification method that does not begin with the aggregated IT2 FS $\tilde{B}$, whereas $\mathrm{TR}_{\mathrm{rew}}(\tilde{B})$ begins with the aggregated IT2 FS $\tilde{B}$.

Greenfield et al. (2009a, b, c) and Greenfield and Chiclana (2011) present an iterative *collapsing method* to defuzzify an IT2 FS $\tilde{B}$ directly. It "generates a T1 embedded set from an IT2 FS that is deemed representative in that it has approximately the same defuzzified value as the original IT2 FS." In the limit, as the number of samples of the primary variable goes to infinity, Greenfield and Chiclana (2011) show, by means of three examples, that the collapsing method approaches $y_{\mathrm{NT}}(\mathbf{x}')$; however, an example that is given in Mendel and Liu (2012) shows that the collapsing method results may be close to $y_{\mathrm{NT}}(\mathbf{x}')$, but may not always equal $y_{\mathrm{NT}}(\mathbf{x}')$. Of course, the collapsing method can only be used if an FOU is available, as is the case only for Centroid type-reduction + defuzzification.

When the input is modeled as an IT2 FS, then a possible alternative to computing the firing interval by means of (9.82) and (9.83), is to compute a similarity between $\tilde{X}_i$ and $\tilde{F}_i^l$ (see, e.g., Bustince 2000; Turksen and Zhong 1990). When the

Jaccard similarity measure is used, then this can be done using the results that are in Exercise 7.46. This approach remains to be explored when it is used in a real-time real-world application.

Computing the similarity between a fuzzy input and its antecedent fuzzy set has found applicability in Computing With Words when the system inputs are restricted to a predetermined and known vocabulary of words that are modeled as IT2 FSs, because all of the similarity computations only need to be performed once ahead of time and can then be stored (Wu and Mendel 2009; Mendel and Wu 2010).

Exercises

9.1 Provide figures that are comparable to Figs. 9.2, 9.3 and 9.12 but using symmetrical triangle FOUs (as in Fig. 6.12a), but when the LMF and UMF are symmetrical). Do this for both minimum and product t-norms.

9.2 Provide figures that are comparable to Figs. 9.5, 9.3 and 9.12 but using symmetrical triangle FOUs (as in Fig. 6.12a, but when the LMF and UMF are symmetrical). Do this for both minimum and product t-norms.

9.3 Provide figures that are comparable to Figs. 9.7, 9.3 and 9.12 but using symmetrical triangle FOUs (as in Fig. 6.12a, but when the LMF and UMF are symmetrical). Do this for both minimum and product t-norms.

9.4 Repeat Example 9.6 using the minimum t-norm.

9.5 Prove that the firing interval is larger in the T1 non-singleton IT2 case than it is in the singleton IT2 case.

9.6 Prove Corollary 9.5.

9.7 Referring to Fig. 9.11c, enumerate the 25 IT2 second-order rule partitions, as was done in Example 9.10 for the five IT2 second-order rule partitions in Fig. 9.11b.

9.8 Obtain formulas that give the total number of IT2 first- and second-order partitions for $X_1 \times X_2 \times X_3$ when each of the three variables is described by three terms whose FOUs are exactly like the ones in Fig. 9.10. Knowing the formulas for the total number of IT2 first- and second-order rule partitions for $X_1 \times X_2$ and $X_1 \times X_2 \times X_3$, can they be generalized to, e.g., $X_1 \times X_2 \times X_3 \times X_4$?

9.9 Repeat Example 9.10 for $X_1 \times X_2$ when each of the two variables is described by three terms whose FOUs are depicted in Fig. 9.19:

(a) Obtain an IT2 first-order rule partition diagram.
(b) How many IT2 first-order rule partitions are there?
(c) In the IT2 first-order rule partitions that have IT2 second-order rule partitions, how many of the latter are there in each such IT2 first-order rule partition?
(d) How many IT2 second-order rule partitions are there in this $X_1 \times X_2$?

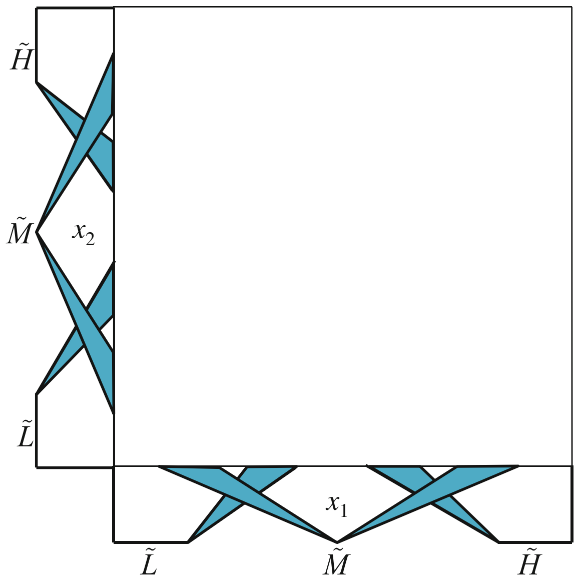


Fig. 9.19 FOUs for Exercise 9.9

9.10 Repeat Example 9.10 for $E \times \dot{E}$ when each of the two variables (error and error-rate, as in an IT2 fuzzy PID controller) is described by two terms (positive and negative) whose FOUs are depicted in Fig. 9.20:

(a) Obtain an IT2 first-order rule partition diagram.
(b) How many IT2 first-order rule partitions are there?
(c) In the IT2 first-order rule partitions that have IT2 second-order rule partitions, how many of the latter are there in each such IT2 first-order rule partition?
(d) How many IT2 second-order rule partitions are there in this $E \times \dot{E}$?

9.11 Repeat Example 9.10 for $X_1 \times X_2$ when each of the two variables is described by three terms whose FOUs are depicted in Fig. 9.21:

(a) Obtain an IT2 first-order rule partition diagram.
(b) How many IT2 first-order rule partitions are there?
(c) In the IT2 first-order rule partitions that have IT2 second-order rule partitions, how many of the latter are there in each such IT2 first-order rule partition?
(d) How many IT2 second-order rule partitions are there in this $X_1 \times X_2$?

9.12 Consider the very special case of an IT2 TSK fuzzy system when its consequent sets are T1 FSs but its antecedents are only T1 FSs. To distinguish this case from the more general one, it is referred to as the A1-C1 case. In this case, the IT2 TSK rules in (9.2) simplify to $(i = 1, \ldots, M)$:

Fig. 9.20 FOUs for Exercise 9.10

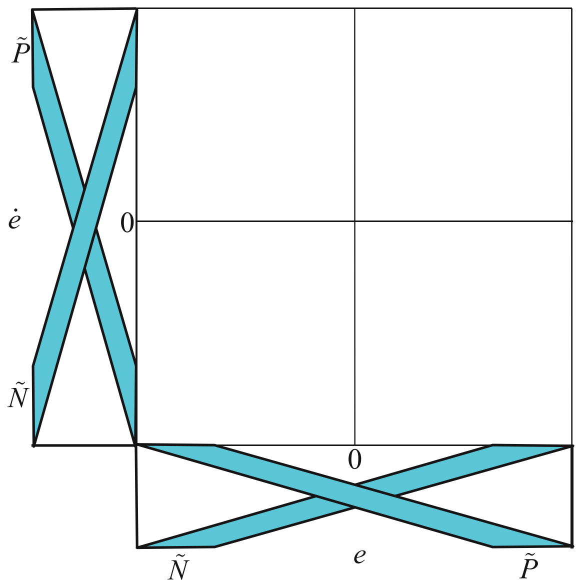
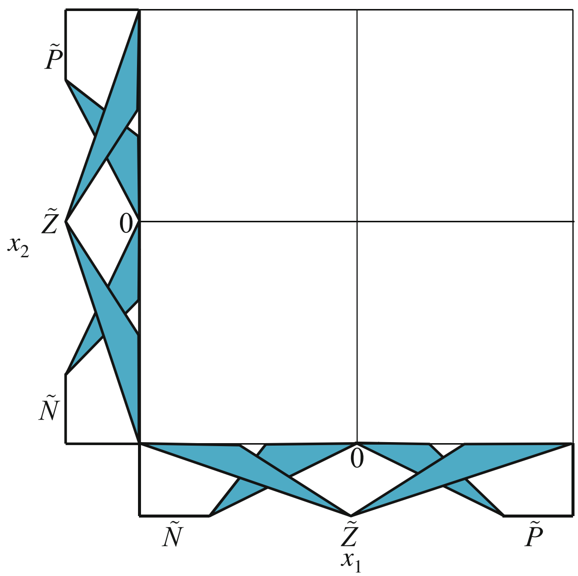


Fig. 9.21 FOUs for Exercise 9.11



$$R^i_{\mathrm{TSK}}: \text{ IF } x_1 \text{ is } F^i_1 \text{ and } \cdots \text{ and } x_p \text{ is } F^i_p, \text{ THEN } Y^i \\ = C^i_0 + C^i_1 x_1 + C^i_2 x_2 + \cdots + C^i_p x_p$$

(a) Derive the following formula for $y^N_{\mathrm{TSK}}(\mathbf{x}')$:

$$y^N_{\mathrm{TSK}}(\mathbf{x}') = \frac{\sum_{i=1}^M f^i(\mathbf{x}')\left(\sum_{k=1}^p c^i_k x'_k + c^i_0\right)}{\sum_{i=1}^M f^i(\mathbf{x}')}$$

where c^i_k is the center of C^i_k, $k = 0, 1, \ldots, p$, and $f^i(\mathbf{x}') = T_{k=1}^p \mu_{F^i_k}(x'_k)$.

(b) Explain why the output of this A1-C1 case is the same as the output of a normalized type-1 TSK fuzzy system whose consequent parameters are the centers of the consequent sets of the A1-C1 IT2 TSK fuzzy system. Consequently, if one is only interested in the defuzzified output of the A1-C1 IT2 TSK fuzzy system, one may just as well use a normalized type-1 TSK fuzzy system directly, because they both provide identical results.

(c) If, however, one is also interested in an uncertainty band about the output, then explain how this information can only be obtained by working with the A1-C1 IT2 TSK fuzzy system.

9.13 Repeat the Sect. 9.7 calculations for $\mathbf{x}' = \mathrm{col}(0.5, -0.5)$.

9.14 Show that when all IT2 MF uncertainties disappear, then WM UB output (9.152) reduces to a correct type-1 defuzzification formula.

9.15 Show that when all IT2 MF uncertainties disappear, then NT outputs (9.155)–(9.157) reduce to correct type-1 defuzzification formulas.

9.16 Show that when all IT2 MF uncertainties disappear, then BMM output (9.158) reduces to a correct type-1 defuzzification formula.

9.17 Explain how an IT2 fuzzy system can be thought of as a collection of a large number of embedded type-1 fuzzy systems.

9.18 Explain why no FBF expansion exists for centroid type-reduction + defuzzification.

9.19 Obtain the IT2 FBF expansion and the IT2 FBFs for IT2 Mamdani fuzzy system with height type-reduction + defuzzification.

9.20 Obtain the IT2 FBF expansion and the IT2 FBFs for the A2-C0 normalized IT2 TSK fuzzy system.

9.21 Obtain the IT2 FBF expansion and the IT2 FBFs for the unnormalized A2-C1 IT2 TSK fuzzy system.

9.22 Obtain the IT2 FBF expansion and the IT2 FBFs for the normalized A2-C1 IT2 TSK fuzzy system.

9.23 Explain whether or not an IT2 FBF expansion exists for a WM UB IT2 fuzzy system.

9.24 Construct a layered architecture figure that is like Fig. 9.16 for:

(a) IT2 TSK fuzzy systems (unnormalized and normalized).
(b) WM UB fuzzy system.
(c) NT fuzzy system.
(d) BMM fuzzy system.

9.25 Referring to Fig. 9.18 (Example 9.17), prove that when $x_1 = 0.1$ and $x_2 = 0.4$ then $y_{\text{COS}}(0.1, 0.4) = 5.5$ and $y_{\text{COS}}(0.1 + \delta, 0.4) = 7$, thereby demonstrating that the corresponding input–output map has a *jump discontinuity* when $x_1 = 0.1$ and $x_2 = 0.4$.

9.26 Here a function approximation example is examined that is different from the usual function approximation examples to which a type-1 fuzzy system is applied. The function to be approximated is $y = 100 - x^2$ where $x \in [-10, 10]$. What is novel about this example is that one only has access to noise-corrupted measurements of y. Consequently, an experiment is created where ten realizations of each measurement are collected. This is done for nine (x, y) pairs. Each of these pairs includes values of y that are corrupted by additive noise that is uniformly distributed in [10, 10]. For each applied input x^i $(i = 1, \ldots, 9)$, one then finds the minimum (y^i_{min}) and the maximum (y^i_{max}) of the 10 y values. The nine $(x^i,\ [y^i_{\text{min}},\ y^i_{\text{max}}])$ pairs that were obtained from this experiment are:

$$
\begin{array}{ll}
(x^1,\ [y^1_{\text{min}},\ y^1_{\text{max}}]) = (-10,\ [-7.79, 6.49]) & (x^6,\ [y^6_{\text{min}},\ y^6_{\text{max}}]) = (2.5,\ [88.02, 103.53]) \\
(x^2,\ [y^2_{\text{min}},\ y^2_{\text{max}}]) = (-7.5,\ [34.72, 52.93]) & (x^7,\ [y^7_{\text{min}},\ y^7_{\text{max}}]) = (5,\ [65.37, 84.32]) \\
(x^3,\ [y^3_{\text{min}},\ y^3_{\text{max}}]) = (-5,\ [66.12, 84.1]) & (x^8,\ [y^8_{\text{min}},\ y^8_{\text{max}}]) = (7.5,\ [34.14, 50.85]) \\
(x^4,\ [y^4_{\text{min}},\ y^4_{\text{max}}]) = (-2.5,\ [84.93, 101.75]) & (x^9,\ [y^9_{\text{min}},\ y^9_{\text{max}}]) = (10,\ [-9.62, 9.62]) \\
(x^5,\ [y^5_{\text{min}},\ y^5_{\text{max}}]) = (0,\ [93.09, 109.95]) &
\end{array}
$$

One rule is formed from each of these pairs, where each of the nine rules is: IF x is A, THEN y is $\tilde{\text{G}}$. Observe that, because only the y-values are uncertain in the given input–output pairs, the antecedent fuzzy sets have been chosen to be T1 FSs and the consequent fuzzy sets have been chosen to be IT2.

In this example, the antecedent fuzzy sets are chosen to be type-1 Gaussians and the consequent sets are chosen to be Gaussian primary MFs but with uncertain means. The centers of the Gaussian antecedent MFs are located at the nine sampled values of x. Each antecedent set has the same standard deviation, arbitrarily chosen to be 1.25. Each consequent set is an IT2 FS that was obtained from a Gaussian primary MF whose mean takes values in the interval $[y^i_{\text{min}},\ y^i_{\text{max}}]$. For illustrative purposes, choose the standard deviation of the nine consequent sets also to be the same, namely 40. In this way, there is a large overlap in the consequent MFs for the nine rules.

The IT2 fuzzy system is Mamdani with singleton fuzzification (since the input x-values can be specified perfectly), maximum t-conorm, product t-norm, product implication, and COS type-reduction + defuzzification. Obtain a formula for the output of this IT2 fuzzy system.

9.27 Convert the ball and beam Exercise 3.15 into one that uses IT2 FSs. Be sure to state the IT2 Zadeh rules, all of the IT2 MFs (you will have to choose them), and do this only for COS type-reduction + defuzzification.

9.28 Convert the truck backing up Exercise 3.16 into one that uses IT2 FSs. Be sure to state the IT2 Relational matrix, all of the IT2 MFs (you will have to choose them), and do this only for COS type-reduction + defuzzification.

9.29 Convert the adaptive learning factor Exercise 3.18 into one that uses IT2 FSs. Be sure to state the IT2 Zadeh rules, all of the IT2 MFs (you will have to choose them), and do this only for COS type-reduction + defuzzification.

9.30 Refer to Sect. 8.3.5 and compute the firing interval for the two fired rules.

References

Aliev, R.A., W. Pedrycz, B.G. Guirimov, R.R. Aliev, U. Ilhan, M. Babagil, and A. Mammadli. 2011. Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization. *Information Sciences* 181: 1591–1608.

Biglarbegian, M., W.W. Melek, and J.M. Mendel. 2008. Stability analysis of type-2 fuzzy systems. In *Proceedings of IEEE FUZZ conference*, Paper # FS0233. Hong Kong, China, June 2008.

Biglarbegian, M., W.W. Melek, and J.M. Mendel. 2010. On the stability of interval type-2 TSK fuzzy logic control systems. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 40: 798–818.

Biglarbegian, M., W.W. Melek, and J.M. Mendel. 2011. On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling. *Information Sciences* 181: 1325–1347.

Bustince, H. 2000. Indicator of inclusion grade for interval-valued fuzzy sets: Applications to approximate reasoning based on interval-valued fuzzy sets. *International Journal of Approximate Reasoning* 23: 137–209.

Castillo, O., J.R. Castro, P. Melin, and A. Rodriguez-Diaz. 2013. Universal approximation of a class of interval type-2 fuzzy neural networks in nonlinear identification. *Advances in Fuzzy Systems* 2013, Article ID 136214: 16 p.

Castro, J.R., O. Castillo, P. Melin, and A. Rodriguez-Diaz. 2009. A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. *Information Sciences* 179: 2175–2193.

Coupland, S., and R.I. John. 2007. Geometric type-1 and type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 15: 3–15.

Du, X., and H. Ying. 2010. Derivation and analysis of the analytical structures of the interval type-2 fuzzy-PI and PD controllers. *IEEE Transactions on Fuzzy Systems* 18 (4): 802–814.

Dziech, A., and M.B. Gorzalczany. 1987. Decision making in signal transmission problems with interval-valued fuzzy sets. *Fuzzy Sets and Systems* 23 (2): 191–203.

Gorzalczany, M.B. 1983. *Interval-valued fuzzy method of approximate inference and its application to the problems of signal transmission and construction of control algorithms*, Ph. D. Thesis, Technical University of Poznan, Poland (in Polish).

Gorzalczany, M.B. 1987. A method of inference in approximate reasoning based on interval-valued fuzzy sets. *Fuzzy Sets and Systems* 21: 1–17.

Greenfield, S., and F, Chiclana. 2011. Type-reduction of the discretised interval type-2 fuzzy set: What happens as discretisation becomes finer? In *Proceedings of 2011 IEEE symposium on advances in type-2 fuzzy logic systems* (T2FUZZ 2011), *April 2011*, pp. 102–109, as part of the IEEE SSCI 2011, Paris, France.

Greenfield, S., F. Chiclana, and R.I. John. 2009a. The collapsing method: Does the direction of collapse affect accuracy? In *Proceedings of IFSA-EUSFLAT*, pp. 980–985. Lisbon, Portugal

Greenfield, S., F. Chiclana, and R.I. John. 2009b. Type-reduction of the discretised interval type-2 fuzzy set. In *Proceedings of IEEE FUZZ Conference*, pp. 738–743, JeJu Island, Korea.

Greenfield, S., F. Chiclana, S. Coupland, and R. John. 2009c. The collapsing method for defuzzification of discretized interval type-2 fuzzy sets. *Information Sciences* 179: 2055–2069.

Hagras, H. 2004. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on Fuzzy Systems* 12 (4): 524–539.

Jamshidi, M. 1997. *Large scale systems: Modeling, control and fuzzy logic*. Upper Saddle River, NJ: Prentice-Hall, PTR (Section 8.3.2, Rule-base reduction).

Juang, C.-F., R.-B. Huang, and Y.-Y. Lin. 2009. A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing. *IEEE Transactions on Fuzzy Systems* 17 (5): 1092–1105.

Juang, C.-F., and K.J. Juang. 2013. Reduced interval type-2 neural fuzzy system using weighted bound-set boundary operation for computation speedup and chip implementation. *IEEE Transactions on Fuzzy Systems* 21: 477–491.

Juang, C.-F., and Y.-W. Tsao. 2008. A type-2 self-organizing neural fuzzy system and its FPGA implementation. *IEEE Transaction. on Systems, Man, and Cybernetics—Part B: Cybernetics* 38 (6).

Juang, C.-F., and P.-H. Wang. 2015. An interval type-2 neural fuzzy classifier learning through soft margin minimization and its human posture classification application. *IEEE Transactions on Fuzzy Systems* 23: 1474–1487.

Kayacan, E., and M.A. Khanesar. 2016. *Fuzzy neural networks for real time control applications*. Amsterdam: Elsevier.

Khanesar, M. A., and J.M. Mendel. 2016. Maclaurin series expansion complexity-reduced center of sets type-reduction + defuzzification for interval type-2 fuzzy systems. In *Proceedings of FUZZ-IEEE 2016, July 2016*, Vancouver, CA.

Kumar, A., S. Sharma, and R. Mitra. 2012. Design of type-2 fuzzy controller based on LQR mapped fusion function. *International Journal of Intelligent Systems and Application* 4 (8): 18–29.

Li, C., J. Yi, and T. Wang. 2011. Stability analysis of SIRMs based on type-2 fuzzy logic control systems. In *Proceedings of FUZZ-IEEE 2011*, pp. 1–7. Taipei, Taiwan.

Liang, Q., and J.M. Mendel. 2000a. Interval type-2 fuzzy logic systems: Theory and design. *IEEE Transactions on Fuzzy Systems* 8: 535–550.

Liang, Q., and J.M. Mendel. 2000b. Designing interval type-2 fuzzy logic systems using an SVD–QR method: Rule reduction. *International Journal of Intelligent Systems* 15: 939–957.

Liang, Q., and J.M. Mendel. 2001. Modeling MPEG VBR video traffic using type-2 fuzzy logic systems. In *Granular computing: An emerging paradigm*, ed. W. Pedrycz, pp. 367–383. Germany: Springer-Verlag, Heidelberg.

Lin, T.-C., and M.-C. Chen. 2011. Adaptive hybrid type-2 intelligent sliding mode control for uncertain nonlinear multivariable dynamical systems. *Fuzzy Sets and Systems* 171: 44–71.

Liu, Z., C.L.P. Chen, and Y. Zhang. 2012. Type-2 hierarchical fuzzy system for high-dimensional data-based modeling with uncertainties. *Soft Computing* 16 (11): 1945–1957.

Lynch C., H. Hagras, and V. Callaghan. 2006. Using uncertainty bounds in the design of embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines. In *Proceedings of IEEE FUZZ conference*, *July 2006*, pp. 7217–7224. Vancouver, CA.

Mendel, J.M. 2001. *Introduction to rule-based fuzzy logic systems*. Upper Saddle River, NJ: Prentice-Hall.

Mendel, J.M., R.I. John, and F. Liu. 2006. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems* 14: 808–821.

Mendel, J.M. 2007. Advances in type-2 fuzzy sets and systems. *Information Sciences* 177: 84–110.

Mendel, J.M. 2013. On KM algorithms for solving type-2 fuzzy set problems. *IEEE Transactions on Fuzzy Systems* 21: 426–446.

Mendel, J. M., and X. Liu. 2012. New closed-form solutions for Karnik-Mendel algorithm + defuzzification of an interval type-2 fuzzy set. In *Proceedings of FUZZ-IEEE 2012*, *June 2012*, pp. 1610–1617. Brisbane, AU.

Mendel, J.M., and X. Liu. 2013. Simplified interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 21 (6): 1056–1069.

Mendel, J.M., H. Hagras, W.-W. Tan, W.W. Melek, and H. Ying. 2014. *Introduction to type-2 fuzzy logic control*. Hoboken, NJ: John Wiley and IEEE Press.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: Aiding people in making subjective judgments"*. Hoboken, NJ: Wiley and IEEE Press.

Nie, M., and W.W. Tan. 2008. Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In *Proceedings of IEEE FUZZ conference*, Paper # FS0339. Hong Kong, China.

Niewiadomski, A., J. Ochelska, and P.S. Szczepaniak. 2006. Interval-valued linguistic summaries of databases. *Control & Cybernetics* 35 (2): 415–443.

Sahab, N., and H. Hagras. 2011. Adaptive non-singleton type-2 fuzzy logic systems: A way forward for handling numerical uncertainties in real world applications.*International Journal of Computers Communications & Control* 6 (3): 503–529.

Sahab, N., and H. Hagras. 2012. Towards comparing adaptive type-2 input based non-singleton type-2 FLS and non-singleton FLSs employing Gaussian inputs. In *Proceedings of FUZZ-IEEE 2012*, WCCI 2012 IEEE *world congress on computational intelligence*, *June 2012*, pp, 1384–1391. Brisbane, AU.

Tao, C.W., J.S. Taur, C.-W. Chang, and Y.-H. Chang. 2012. Simplified type-2 fuzzy sliding controller for wing rock system. *Fuzzy Sets Systems* 207: 111–129.

Türksen, I.B., and Z. Zhong. 1990. An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets. *Fuzzy Sets and Systems* 34: 323–346.

Wu, D. 2011. An interval type-2 fuzzy logic system cannot be implemented by traditional type-1 fuzzy logic systems. In *Proceedings of world conference on soft computing, May 2011*. San Francisco, CA.

Wu, D. 2012. On the fundamental differences between interval type-2 and type-1 fuzzy logic controllers. *IEEE Transactions on Fuzzy Systems* 20: 832–848.

Wu, D. 2013a. Approaches for reducing the computational costs of interval type-2 fuzzy logic systems: Overview and comparisons. *IEEE Transactions on Fuzzy Systems* 21 (1) 80–93

Wu, D. 2013b. Two differences between interval-type-2 and type-1 fuzzy logic controllers: Adaptiveness and novelty. In *Advances in type-2 fuzzy sets and systems: Theory and applications,* ed. A. Sadeghian, J.M. Mendel, and H. Tahayori, pp. 33–48. Springer, New York.

Wu, D., and J.M. Mendel. 2009. Perceptual reasoning for perceptual computing: A similarity-based approach. *IEEE Transactions on Fuzzy Systems* 17: 1397–1411.

Wu, D., and J.M. Mendel. 2011. On the continuity of type-1 and interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 19 (1): 179–192.

Wu, D., and W. W. Tan. 2004. A type-2 fuzzy logic controller for the liquid-level process. In *Proceedings of IEEE FUZZ conference, July 2004*, vol. 2, 953–958. Budapest, Hungary.

Wu, D., and W.W. Tan. 2005. Computationally efficient type-reduction strategies for a type-2 fuzzy logic controller. In *Proceedings of IEEE FUZZ Conference, May 2005*, pp. 353–358. Reno, NV.

Wu, H., and J.M. Mendel. 2002. Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 10: 622–639.

Ying, H. 2008. General interval type-2 Mamdani fuzzy systems are universal approximators. In *Proceedings of NAFIPS 2008*, Paper #978-1-4244-2352-1/08. New York City, NY.

Ying, H. 2009. Interval type-2 Takagi-Sugeno fuzzy systems with linear rule consequents are universal approximators. *Proceedings of NAFIPS 2009*, Paper #978-1-4244-4577-6/09. Cincinnati, OH.

Zadeh, L.A. 1996. Fuzzy logic = Computing with words. *IEEE Transactions on Fuzzy Systems* 4: 103–111.

Zhou, S.-M., J.M. Garibaldi, R.I. John, and F. Chiclana. 2009. On constructing parsimonious type-2 fuzzy logic systems via influential rule selection. *IEEE Transactions on Fuzzy Systems* 17: 654–667.

Chapter 10
Interval Type-2 Fuzzy Systems: Design Methods and Applications

10.1 Designing IT2 Fuzzy Systems

From the detailed discussions given about the five elements that comprise the Fig. 9.1a, or the four elements that comprise Fig. 9.1b IT2 fuzzy systems, it should be clear that there are many possibilities to choose from. Table 10.1 enumerates them. Not included in that table is which input variables should be used, because this is application dependent. As stated in Sect. 4.1:

> For some applications the choice of which variables to use is easy because of historical precedence (e.g., for fuzzy logic control one would choose the system's states). For others, the choice may be very challenging because there may be no historical precedence (e.g., for time-series forecasting one would choose delayed versions of the main variable, however, how many delays to choose is usually unknown and may need to be varied). In the latter situation, it is best to seek the advice of experts, or, if none are available, to create more than one fuzzy system in order to see which choices lead to a best design.

Although Table 10.1 should be self-explanatory, two examples are presented next that illustrate its use.

Example 10.1 Table 10.2 is for an IT2 Mamdani fuzzy system which uses singleton fuzzification, product t-norm, COS type-reduction + D (so fired rule output sets are not combined, and approximate TR + D and direct D are not applicable), four input variables, three terms for each variable all of which are named *low*, *moderate* and *high*, four output terms named *none to very little*, *some*, *a moderate amount* and *a lot*, 81 rules (3^4), and all FOUs are like the ones in Fig. 6.17 (Gaussian primary MFs with uncertain means; the Gaussians will be truncated for the leftmost and rightmost words) whose parameters will be optimized.

Example 10.2 Table 10.3 is for an A2-C0 IT2 TSK fuzzy system which uses IT2 non-singleton fuzzification, minimum t-norm, normalized type-reduction + D, three input variables, three terms for each variable all of which are named *light*, *moderate,* and *heavy*, a consequent function for each rule that is a crisp linear

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_10

Table 10.1 Choices that need to be made to specify or design an IT2 fuzzy system

Choices for	Kind of IT2 fuzzy system							
• Rules	Mamdani				TSK			
					A2-C0		A2-C1	
• Fuzzifier	Singleton	T1 non-singleton	IT2 non-singleton		Singleton	T1 non-singleton	IT2 non-singleton	
• t-norm	Minimum		Product		Minimum		Product	
• Combining fired rule output sets	Union		None		NA			
• Type-reduction + D	Centroid	Height	COS	NA	Unnormalized		Normalized	
• Approximate TR +D	WM UB		NA		NA			
• Direct D	NT	BMM	NA		NA			
• Number of input variables	p				p			
• Number and names of terms per input	$\{\tilde{X}_{1j}\}_{j=1}^{Q_1}$	$\{\tilde{X}_{2j}\}_{j=1}^{Q_2}$	$\cdots$	$\{\tilde{X}_{pj}\}_{j=1}^{Q_p}$	$\{\tilde{X}_{1j}\}_{j=1}^{Q_1}$	$\{\tilde{X}_{2j}\}_{j=1}^{Q_2}$	$\cdots$	$\{\tilde{X}_{pj}\}_{j=1}^{Q_p}$
• Number and names of terms for, or structure of, output	$\{\tilde{Y}_j\}_{j=1}^{Q_y}$				$\{g^l(\mathbf{x})\}_{l=1}^{M}$			
• Number of rules	M				M			
• Kind of FOU[a]	Specify it		NA		Specify it		NA	
• Kind of UMFs	Triangle	Trapezoid	Other	NA	Triangle	Trapezoid	Other	NA
• Kind of LMFs	Triangle	Trapezoid	Other	NA	Triangle	Trapezoid	Other	NA
• FOU parameters	Pre-specified		Optimized		Pre-specified		Optimized	

TR denotes type-reduction and *D* denotes defuzzification

[a]If "Kind of FOU" is specified, then "Kind of UMFs" and "Kind of LMFs " must be specified as "NA," or if "Kind of FOU" is "NA" then both "Kind of LMF" and "Kind of UMF" must be specified

Table 10.2 Example of choices (shaded) made to specify or design an IT2 Mamdani fuzzy system

Choices for	Kind of IT2 fuzzy system							
• Rules	Mamdani				TSK			
					A2-C0		A2-C1	
• Fuzzifier	Singleton	T1 non-singleton	IT2 non-singleton		Singleton	T1 non-singleton	IT2 non-singleton	
• t-norm	Minimum		Product		Minimum		Product	
• Combining fired rule output sets	Union		None		NA			
• Type-reduction + D	Centroid	Height	COS	NA	Unnormalized		Normalized	
• Approximate TR +D	WM UB		NA		NA			
• Direct D	NT	BMM	NA		NA			
• Number of input variables	4				p			
• Number and names of terms per input	*low*	*moderate*	*high*		$\{\tilde{X}_{1j}\}_{j=1}^{Q_1}$	$\{\tilde{X}_{2j}\}_{j=1}^{Q_2}$	$\cdots$	$\{\tilde{X}_{pj}\}_{j=1}^{Q_p}$
• Number and names of terms for, or structure of, output	*none to very little*	*some*	*a moderate amount*	*a lot*	$\{g^l(\mathbf{x})\}_{l=1}^{M}$			
• Number of rules	81				M			
• Kind of FOU[a]	Gaussian with uncertain mean		NA		Specify it		NA	
• Kind of UMFs	Triangle	Trapezoid	Other	NA	Triangle	Trapezoid	Other	NA
• Kind of LMFs	Triangle	Trapezoid	Other	NA	Triangle	Trapezoid	Other	NA
• FOU parameters	Pre-specified		Optimized		Pre-specified		Optimized	

TR denotes type-reduction and *D* denotes defuzzification

[a]If "Kind of FOU" is specified, then "Kind of UMFs" and "Kind of LMFs " must be specified as "NA," or if "Kind of FOU" is "NA" then both "Kind of LMF" and "Kind of UMF" must be specified

Table 10.3 Example of choices (shaded) made to specify or design an IT2 TSK fuzzy system

Choices for	Kind of IT2 fuzzy system: Mamdani	Kind of IT2 fuzzy system: TSK
• Rules	Mamdani	TSK: A2-C0 / A2-C1
• Fuzzifier	Singleton / T1 non-singleton / IT2 non-singleton	Singleton / T1 non-singleton / IT2 non-singleton
• t-norm	Minimum / Product	Minimum / Product
• Combining fired rule output sets	Union / None	NA
• Type-reduction + D	Centroid / Height / COS / NA	Unnormalized / Normalized
• Approximate TR +D	WM UB / NA	NA
• Direct D	NT / BMM / NA	NA
• Number of input variables	p	3
• Number and names of terms per input	$\{\tilde{X}_{1j}\}_{j=1}^{Q_1}$ $\{\tilde{X}_{2j}\}_{j=1}^{Q_2}$ $\cdots$ $\{\tilde{X}_{pj}\}_{j=1}^{Q_p}$	*light* / *moderate* / *heavy*
• Number and names of terms for, or structure of, output	$\{\tilde{Y}_j\}_{j=1}^{Q_y}$	$\{c_0^l + c_1^l x_1 + c_2^l x_2 + c_3^l x_3\}_{l=1}^{M}$
• Number of rules	M	27
• Kind of FOU[a]	Specify it / NA	Specify it / NA
• Kind of UMFs	Triangle / Trapezoid / Other / NA	Triangle / Trapezoid / Other / NA
• Kind of LMFs	Triangle / Trapezoid / Other / NA	Triangle / Trapezoid / Other / NA
• FOU parameters	Pre-specified / Optimized	Pre-specified / Optimized

TR denotes type-reduction and *D* denotes defuzzification

[a]If "Kind of FOU" is specified, then "Kind of UMFs" and "Kind of LMFs " must be specified as "NA," or if "Kind of FOU" is "NA" then both "Kind of LMF" and "Kind of UMF" must be specified

combination of the three input variables, 27 rules (3^3), normal triangle UMF and LMFs (the apex of the triangles will occur at 0 for *light* and at $\max x_3$ for *heavy*) all of whose free parameters are pre-specified.

Ultimately, after one has made all of the choices indicated in Table 10.1, one must fix the parameters of the FOUs. Either they are pre-specified or tuned just as the MFs in a T1 fuzzy system are either pre-specified or tuned.

There exists a multitude of design methods that can be used to construct IT2 fuzzy systems and that have different properties and characteristics. As was the situation for the design of a T1 fuzzy system, some of these design methods are data-intensive, some are aimed at computational simplicity, some are recursive (thus giving the fuzzy system an adaptive nature), some are offline, and some are application-specific.

The goal of this section is *not* to describe the different design approaches in complete detail. Instead, some of them are described briefly in connection with the following three problems:

1. Given N input–output numerical data *training* pairs, $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)})$, …, $(\mathbf{x}^{(N)} : y^{(N)})$, where $\mathbf{x}$ is the vector input and y is the scalar output of a singleton IT2 fuzzy system, completely specify a singleton Mamdani or TSK IT2 fuzzy system using the training data.
2. Given N input–output numerical data *training* pairs, $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots$, $(\mathbf{x}^{(N)} : y^{(N)})$, where $\mathbf{x}$ is the vector *noisy* input and y is the scalar (possibly) *noisy*

Table 10.4 Firing interval computations ($s = 1, \ldots, M^*$)

Fuzzifier	$\underline{f}^s(\mathbf{x}')$	$\bar{f}^s(\mathbf{x}')$
Singleton	$T_{i=1}^p \underline{\mu}_{F_i^s}(x_i')$	$T_{i=1}^p \bar{\mu}_{F_i^s}(x_i')$
T1 non-singleton	$T_{i=1}^p \max_{x_i \in X_i} \mu_{X_i}(x_i \vert x_i') \bigstar \underline{\mu}_{\tilde{F}_i^s}(x_i)$	$T_{i=1}^p \max_{x_i \in X_i} \mu_{X_i}(x_i \vert x_i') \bigstar \bar{\mu}_{\tilde{F}_i^s}(x_i)$
IT2 non-singleton	$T_{i=1}^p \max_{x_i \in X_i} \underline{\mu}_{X_i}(x_i \vert x_i') \bigstar \underline{\mu}_{\tilde{F}_i^s}(x_i)$	$T_{i=1}^p \max_{x_i \in X_i} \bar{\mu}_{\tilde{X}_i}(x_i \vert x_i') \bigstar \bar{\mu}_{\tilde{F}_i^s}(x_i)$

output of a T1 non-singleton IT2 fuzzy system, completely specify a T1 non-singleton Mamdani or TSK IT2 fuzzy system using the training data.

3. Given N input–output numerical data *training* pairs, $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots, (\mathbf{x}^{(N)} : y^{(N)})$, where $\mathbf{x}$ is the vector *noisy* input and y is the scalar (possibly) *noisy* output of an IT2 non-singleton fuzzy system, completely specify an IT2 non-singleton Mamdani or TSK fuzzy system using the training data.

For illustrative purposes, *all designs in this chapter assume Gaussian primary MFs with uncertain standard deviations* (as in Fig. 6.16). The FOU that is associated with these primary MFs is described by three design parameters, $\{m, \sigma_1, \sigma_2\}$. Common to the designs of all IT2 fuzzy systems are the firing interval computations, which are collected in Table 10.4 for the convenience of the readers.

Except for an IT2 Mamdani fuzzy system that uses centroid type-reduction + defuzzification, the outputs of all other IT2 fuzzy systems can be expressed as some sort of IT2 FBF expansion (Sect. 9.12), namely:

$$y(\mathbf{x}) = \sum_{s=1}^{M^*} \lambda^s \phi_s(\mathbf{x}) \tag{10.1}$$

In (10.1), the value of M^* and the nature of the dependence of the IT2 FBSs on $\underline{f}^s(\mathbf{x})$ and $\bar{f}^s(\mathbf{x})$ depend on the specific IT2 fuzzy system [e.g., see Examples 9.13–9.16 to figure out what M^* is for each IT2 fuzzy system, and what the exact dependence of the FBFs are on $\underline{f}^s(\mathbf{x})$ and $\bar{f}^s(\mathbf{x})$]. To keep things as general as possible, the IT2 FBFs will be expressed here as:

$$\phi_s(\mathbf{x}) = \phi_s[\{\underline{f}^j(\mathbf{x})\}_{j=1}^M, \{\bar{f}^j(\mathbf{x})\}_{j=1}^M, \mathbf{x}] \tag{10.2}$$

It is only for an IT2 TSK fuzzy system that $\phi_s(\mathbf{x})$ depends explicitly on $\mathbf{x}$ [e.g., see (9.179)], and for some IT2 fuzzy systems $\phi_s(\mathbf{x})$ depends on both $\{\underline{f}^j(\mathbf{x})\}_{j=1}^M$ and $\{\bar{f}^j(\mathbf{x})\}_{j=1}^M$ [e.g., see (9.173), (9.174), (9.179) and (9.181)], whereas for others it depends on just $\{\underline{f}^j(\mathbf{x})\}_{j=1}^M$ or $\{\bar{f}^j(\mathbf{x})\}_{j=1}^M$ [e.g., see (9.184) and (9.185)].

Each design method establishes how to specify the IT2 fuzzy system's FOU and consequent parameters using the training pairs $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots, (\mathbf{x}^{(N)} : y^{(N)})$. Before describing some design methods, it is important to relate the number of rules, M, and the number of training samples, N, to one another.

If no tuning of the IT2 fuzzy system's parameters is done, then there can be as many as $M = N$ rules. If tuning is used, and one abides by the commonly used design principle that there must be fewer tuned design parameters than training pairs, then $M < N$. The exact inequality relationship between M and N depends on many choices, as are illustrated in Tables 10.2 and 10.3. Because there are so many different kinds of IT2 fuzzy systems, the next four examples provide the inequality relationship between M and N for the IT2 fuzzy systems that have been described in Examples 9.13–9.16, in Sect. 9.12.

Example 10.3 (**IT2 Mamdani fuzzy system with COS type-reduction + defuzzification**) Each IT2 Zadeh rule has p antecedent IT2 FSs, $\tilde{F}_i^l$, each of which is described by an FOU with three design parameters $\{m_{\tilde{F}_i^s}, \sigma^1_{\tilde{F}_i^s}, \sigma^2_{\tilde{F}_i^s}\}_{i=1}^p$; hence, each rule contributes $3p$ antecedent parameters. For COS type-reduction (Sect. 8.3.4), the IT2 consequent FS can be replaced by the domain of its centroid, $[c_l(\tilde{G}^s), c_r(\tilde{G}^s)] \equiv [c_l^s, c_r^s]$; hence, each rule contributes 2 consequent parameters. Singleton fuzzification contributes no additional design parameter; T1 non-singleton fuzzification contributes one additional design parameter (the standard deviation of the Gaussian T1 fuzzy number) and IT2 non-singleton fuzzification contributes two additional design parameters (the standard deviations of the Gaussian lower and upper MFs—T1 fuzzy numbers). The total numbers of design parameters are:

$$\begin{cases} (3p+2)M & \text{singleton fuzzifier} \\ (3p+2)M+p & \text{T1 non-singleton fuzzifier} \\ (3p+2)M+2p & \text{IT2 non-singleton fuzzifier} \end{cases} \tag{10.3}$$

Requiring the total number of design parameters to be less than $N = N'$, means:

$$\begin{cases} M < N'/(3p+2) & \text{singleton fuzzifier} \\ M < (N'-p)/(3p+2) & \text{T1 non-singleton fuzzifier} \\ M < (N'-2p)/(3p+2) & \text{IT2 non-singleton fuzzifier} \end{cases} \tag{10.4}$$

Equation (10.4) constrains the number of rules that can be used. Because M must be an integer, one chooses $M = M'$ as an integer that is smaller than the right-hand sides of the inequalities that are given in (10.4). The same is true for the results in (10.6), (10.8), and (10.10) below.

Example 10.4 (**A2-C0 unnormalized IT2 TSK fuzzy system**) The number of antecedent design parameters for an IT2 TSK rule is the same as for an IT2 Zadeh rule, namely (see Example 10.3) $3p$. The consequent of each IT2 TSK rule has $p+1$ parameters; hence, each rule contributes $4p+1$ design parameters. Again, singleton fuzzification contributes no additional design parameter; T1 non-singleton fuzzification contributes one additional design parameter (the standard deviation of the Gaussian T1 fuzzy number) and IT2 non-singleton fuzzification contributes two additional design parameters (the standard deviations of the Gaussian lower and upper MFs—T1 fuzzy numbers). The total numbers of design parameters are:

$$\begin{cases} (4p+1)M & \text{singleton fuzzifier} \\ (4p+1)M+p & \text{T1 non-singleton fuzzifier} \\ (4p+1)M+2p & \text{IT2 non-singleton fuzzifier} \end{cases} \tag{10.5}$$

Requiring the total number of design parameters to be less than $N = N'$, means:

$$\begin{cases} M < N'/(4p+1) & \text{singleton fuzzifier} \\ M < (N'-p)/(4p+1) & \text{T1 non-singleton fuzzifier} \\ M < (N'-2p)/(4p+1) & \text{IT2 non-singleton fuzzifier} \end{cases} \tag{10.6}$$

Example 10.5 (**NT direct defuzzification fuzzy system**) Because this fuzzy system is for an IT2 Zadeh rule, it has the same number of antecedent design parameters as an IT2 Mamdani fuzzy system with COS type-reduction + defuzzification (Example 10.3), namely $3p$. Examining (9.180), observe that there are also M coefficients $\bar{y}^i$. Again, singleton fuzzification contributes no additional design parameter; T1 non-singleton fuzzification contributes one additional design parameter (the standard deviation of the Gaussian T1 fuzzy number) and IT2 non-singleton fuzzification contributes two additional design parameters (the standard deviations of the Gaussian lower and upper MFs—T1 fuzzy numbers). The total numbers of design parameters are:

$$\begin{cases} (3p+1)M & \text{singleton fuzzifier} \\ (3p+1)M+p & \text{T1 non-singleton fuzzifier} \\ (3p+1)M+2p & \text{IT2 non-singleton fuzzifier} \end{cases} \tag{10.7}$$

Requiring the total number of design parameters to be less than $N = N'$, means:

$$\begin{cases} M < N'/(3p+1) & \text{singleton fuzzifier} \\ M < (N'-p)/(3p+1) & \text{T1 non-singleton fuzzifier} \\ M < (N'-2p)/(3p+1) & \text{IT2 non-singleton fuzzifier} \end{cases} \tag{10.8}$$

Example 10.6 (**BMM direct defuzzification fuzzy system**) This fuzzy system also has the same number of antecedent design parameters as an IT2 Mamdani fuzzy system with COS type-reduction + defuzzification (Example 10.3), namely $3p$. Examining (9.182), observe that there are also M coefficients y_i and two scaling parameters, m and n. Again, singleton fuzzification contributes no additional design parameter; T1 non-singleton fuzzification contributes one additional design parameter (the standard deviation of the Gaussian T1 fuzzy number) and IT2 non-singleton fuzzification contributes two additional design parameters (the standard deviations of the Gaussian lower and upper MFs—T1 fuzzy numbers). The total numbers of design parameters are:

$$\begin{cases} (3p+1)M+2 & \text{singleton fuzzifier} \\ (3p+1)M+2+p & \text{T1 non-singleton fuzzifier} \\ (3p+1)M+2+2p & \text{IT2 non-singleton fuzzifier} \end{cases} \tag{10.9}$$

Requiring the total number of design parameters to be less than $N = N'$, means:

$$\begin{cases} M < (N'-2)/(3p+1) & \text{singleton fuzzifier} \\ M < (N'-2-p)/(3p+1) & \text{T1 non-singleton fuzzifier} \\ M < (N'-2-p)/(3p+1) & \text{IT2 non-singleton fuzzifier} \end{cases} \tag{10.10}$$

At a very high level, one can think of a multitude of designs associated with the preceding formulations. In some formulations, FOUs may be pre-specified in which case the IT2 FBFs in (10.1) are known and only its $M^*\lambda$-coefficients have to be tuned using the training data. In this case, the IT2 fuzzy system is linear in these coefficients and their tuning by means of optimization is relatively easy.

In other formulations only the FOU shapes are pre-specified but their parameters have to be tuned using the training data. Because the IT2 FBFs in (10.1) are nonlinear functions of the antecedent and fuzzifier-model parameters, their tuning by means of optimization is more difficult. How the $M^*\lambda$-coefficients are tuned in these other formulations varies. In one approach, (10.1) is viewed as nonlinear in all design parameters, including the $M^*\lambda$-coefficients, because they multiply the IT2 FBFs. In another approach, the $M^*\lambda$-coefficients are still treated as linear coefficients in (10.1) but only after the IT2 FBFs are specified. In the latter approach, parameter tuning occurs in two iterative stages: (1) Tune the IT2 FBF parameters, and then (2) tune the $M^*\lambda$-coefficients.

Naturally, other even more complicated design problems can be formulated that try to answer questions such as: (a) What shape FOU LMFs and UMFs should be used? (b) How many and which antecedents should be used (e.g., can p be reduced?)? and, (c) How many and which rules should be used (e.g., can M be reduced?)? As mentioned in Sect. 4.1, evolutionary optimization algorithms can be used to address the first of these questions, systematic trial and error or evolutionary optimization algorithms are viable approaches to answering the second question, and singular value decomposition techniques and evolutionary optimization algorithms can be used to answer the third question. Because answers to the second and third questions can reduce the complexity of the fuzzy system, these questions should not be ignored.

Before briefly touching on some design methods, note that there can be two very different *approaches to the tuning of a singleton IT2 fuzzy system*:

1. **Partially dependent approach**: In this approach, one first designs the best possible T1 fuzzy system by tuning all of its parameters, and then uses these parameters in some way to *initialize* the parameters of the singleton IT2 fuzzy system.
2. **Totally independent approach**: In this approach, all of the parameters of the singleton IT2 fuzzy system are tuned without the benefit of a previous type-1

design. In this design the parameters of the singleton IT2 fuzzy system are usually initialized in a random manner.

In addition, there can be two very different *approaches to the tuning of a T1 non-singleton IT2 fuzzy system*:

1. **Partially dependent approach**: In this approach, one first designs the best possible singleton IT2 fuzzy system, by tuning all of its parameters, and then updates the design by: (a) keeping all of the parameters that are shared by the singleton and T1 non-singleton IT2 fuzzy systems fixed at the values obtained from the best possible singleton IT2 fuzzy system; and, (b) tuning only the new parameter(s) of the T1 non-singleton IT2 fuzzy system. In the present case, only the standard deviations, σ_{X_k}, would be tuned.
2. **Totally independent approach**: In this approach, all of the parameters of the T1 non-singleton IT2 fuzzy system are tuned. If, perchance, a singleton IT2 fuzzy system has already been designed, then its parameters can be used as the initial parameters for the tuning algorithms of the parameters that are shared by the singleton and T1 non-singleton IT2 fuzzy systems.

Finally, there can be two very different *approaches to the tuning of an IT2 non-singleton IT2 fuzzy system*:

1. **Partially dependent approach**: In this approach, one first designs the best possible T1 non-singleton IT2 fuzzy system, by tuning all of its parameters, and then updates the design by: (a) keeping all of the parameters that are shared by the T1 non-singleton and IT2 non-singleton IT2 fuzzy systems fixed at the values obtained from the best possible T1 non-singleton IT2 fuzzy system; and (b) tuning only the new parameter(s) of the IT2 non-singleton IT2 fuzzy system. In the present case, only the standard deviations, σ_{k1}^l and σ_{k2}^l, would be tuned, but they can be initialized by using σ_{X_k} obtained from the T1 non-singleton design.
2. **Totally independent approach**: In this approach, all of the parameters of the IT2 non-singleton IT2 fuzzy system are tuned. If, perchance, a T1 non-singleton IT2 fuzzy system has already been designed, then its parameters can be used as the initial parameters for the tuning algorithms of the parameters that are shared by the T1 non-singleton and IT2 non-singleton IT2 fuzzy systems.

One would expect the best performance to be obtained by the totally independent approach; however, sometimes it is very useful to use the partially dependent approach to observe the incremental improvement that can be obtained from the previous design to the new design.

10.2 Some Design Methods

This section briefly describes some design methods for IT2 fuzzy systems. Some involve no tuning of FOU (or MF) parameters and others involve lots of tuning—optimization—of those parameters.

10.2.1 IT2 WM Method

Here, it is explained how the one-pass WM Method that was described in Sect. 4.2.1.2 can be modified for an IT2 fuzzy system. Recall that the WM Method generates a set of IF–THEN rules by using the given training data one time, and then combines the rules in a common rule-base to construct a final fuzzy system. Given a set of training data—input–output pairs—

$$(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots, (\mathbf{x}^{(N)} : y^{(N)}) = \left\{ x_1^{(t)}, x_2^{(t)}, \ldots, x_p^{(t)} : y^{(t)} \right\}_{t=1}^{N} \quad (10.11)$$

where $x_1, x_2, \ldots, x_p$ are inputs and y is the output, one proceeds as follows to construct an IT2 fuzzy system:

1. Let $[X_1^-, X_1^+], [X_2^-, X_2^+], \ldots, [X_p^-, X_p^+]$, $[Y^-, Y^+]$ be the domain intervals of the input and output variables, respectively, where domain interval implies the interval the variable is most likely to lie in. Each domain interval is divided into $2L+1$ regions, where L can be different for each variable. Then, FOUs are assigned to the regions, labeled $\tilde{SL}$ (Small L), …, $\tilde{S}1$ (Small 1), $\tilde{CE}$ (Center), $\tilde{B}1$ (Big 1), …, and $\tilde{BL}$ (Big L). Of course, other label names can be used instead of these names, and the choice of their FOUs should be related to a careful understanding of the application.
2. Because of overlapping FOUs (e.g., see Fig. 6.13), it frequently happens that $x_k^{(t)}$ is in more than one IT2 FS. Therefore, the membership of each input–output point is evaluated in regions where it may occur, and the given $x_1^{(t)}, x_2^{(t)}, \ldots, x_p^{(t)}$, or $y^{(t)}$ are assigned to the region with maximum membership. This is done using the material in Sect. 6.9, specifically (6.64), reexpressed here as[1] ($t = 1, \ldots, N$ and $k = 1, \ldots, p$):

$$L(x_k^{(t)}) = \arg\ \max_{\forall \tilde{F}_i} [u^{cg}_{\tilde{F}_1(x_k^{(t)})}, u^{cg}_{\tilde{F}_2(x_k^{(t)})}, \ldots, u^{cg}_{\tilde{F}_p(x_k^{(t)})}] \quad (10.12)$$

For IT2 FSs, a formula for $u^{cg}_{\tilde{F}_j(x_k^{(t)})}$ is (6.65).

[1]See Sect. 6.9 for a definition of the quantities in (10.12).

3. To resolve conflicting rules (rules with the same antecedent FOUs and different consequent FOUs), a degree is assigned to each rule as follows: Let $(t = 1,\ldots,N$ and $k = 1,\ldots,p)$

$$\max_{\forall \tilde{F}_i}[u^{cg}_{\tilde{F}_1(x_k^{(t)})}, u^{cg}_{\tilde{F}_2(x_k^{(t)})}, \ldots, u^{cg}_{\tilde{F}_p(x_k^{(t)})}] \equiv \zeta(x_k^{(t)}) \tag{10.13}$$

   with a comparable definition for $y^{(t)}$, namely $\zeta(y^{(t)})$. Then, the *degree for the* tth *IT2 Zadeh rule*, $\tilde{R}^t_Z$, is defined as $(t = 1,\ldots,N)$:

$$D(\tilde{R}^t_Z) = \left[\prod_{k=1}^{p} \zeta(x_k^{(t)})\right] \cdot \zeta(y^{(t)}) \tag{10.14}$$

4. In the event of conflicting rules, the rule with the highest degree is kept in the rule-base, and all other conflicting rules are discarded.

These steps extract the rules from the data and incorporate antecedent and consequent uncertainties into this process. Knowing the antecedent and consequent IT2 FOUs for all rules, it is straightforward to translate this information into a completely described IT2 fuzzy system. Note that all of the parameters of this IT2 fuzzy system have been specified as part of this one-pass design.

This method can be applied to noisy or nose-free data. If it is applied to noisy data then one should use either a T1 or an IT2 non-singleton fuzzifier. For the former, one must determine values for $\sigma_{X_k}(k = 1,\ldots,p)$. For the latter, one must determine values for $\sigma^1_{X_k}$ and $\sigma^2_{X_k}$ $(k = 1,\ldots,p)$. This requires the variance (or a range on the variance) of the additive measurement noise on each measured input be known ahead of time (which is not very likely), or that it (they) can be estimated from the data; otherwise, one must also use another design method in which the variance parameters are tuned.

Exercises 10.3 and 10.4 ask the reader to apply the IT2 WM method to the same data that were used in Example 4.4, and to compare each found IT2 Zadeh rule with the Example 4.4 T1 Zadeh rule.

As was true for a T1 fuzzy system, although the IT2 WM Method is simple, it does have some shortcomings, namely: (1) how to choose the parameters of the antecedent and consequent FOUs is left as an open issue, and (2) it can lead to an IT2 fuzzy system that still has too many rules.

10.2.2 Least-Squares Method

Using the notation for the elements in the training set, (10.1) can be expressed as $(t = 1,\ldots,N)$:

$$y(\mathbf{x}^{(t)}) = \sum_{s=1}^{M^*} \lambda^s \phi_s(\mathbf{x}^{(t)}) \tag{10.15}$$

When all of the antecedent parameters, in the IT2 FBFs, are fixed ahead of time by the designer then only the M^* λ^s-parameters in (10.1) are unknown, and it would seem that they could be found by using the least-squares method that was described in Sect. 4.2.2. For this to be true it must be possible to compute the elements in the IT2 FBF matrix Φ:

$$\Phi = \begin{pmatrix} \phi_1(\mathbf{x}^{(1)}) & \cdots & \phi_{M^*}(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}^{(N)}) & \cdots & \phi_{M^*}(\mathbf{x}^{(N)}) \end{pmatrix} \tag{10.16}$$

Example 10.7 Let us examine whether or not (10.16) can be computed for the IT2 FBFs in Examples 9.13–9.16, beginning with Example 9.16 and working back to Example 9.13.

- **BMM direct defuzzification fuzzy system**: The IT2 FBFs for this system are given in (9.184) and (9.185). Given $\mathbf{x} = \mathbf{x}'$, then all M $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$ can be computed by using the formulas that are in Table 10.4. Hence, (10.16) can be computed for this IT2 fuzzy system.
- **NT direct defuzzification fuzzy system**: The IT2 FBFs for this system are given in (9.181). Because they also only depend upon all M $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$, (10.16) can also be computed for this IT2 fuzzy system.
- **A2-C0 unnormalized IT2 TSK fuzzy system**: The IT2 FBFs for this system are given in (9.179). Because they also only depend upon all M $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$, as well as $\mathbf{x}'$, which is given, (10.16) can also be computed for this IT2 fuzzy system.
- **IT2 Mamdani fuzzy system with COS type-reduction + defuzzification**: The IT2 FBFs for this system are given in (9.173) and (9.174); however, not only do they depend upon all M $\underline{f}^i(\mathbf{x}')$ and $\bar{f}^i(\mathbf{x}')$, but they also depend upon the two switching functions $(i = 1, \ldots, M)$ δ_l^i and δ_r^i, and therein lies the rub. The problem is that, to know the *2 M IT2* FBFs $\phi_l^i(\mathbf{x})$ and $\phi_r^i(\mathbf{x})$ one *first* needs to know all 2 M $c_l(\tilde{G}^i)$ and $c_r(\tilde{G}^i)$ [which become the λ^s in (10.15)], because prior to the very first step of, e.g., using the EIASC or EKM algorithms to perform COS type-reduction, all M $c_l(\tilde{G}^i)$ must be rank-ordered and all M $c_r(\tilde{G}^i)$ must also be rank-ordered. But, if numerical values are not known ahead of time for all 2 M $c_l(\tilde{G}^i)$ and $c_r(\tilde{G}^i)$, it is impossible to do this; hence, the *2 M IT2* FBFs $\phi_l^i(\mathbf{x})$ and $\phi_r^i(\mathbf{x})$ cannot be computed, which means that (10.16) cannot be computed for this IT2 fuzzy system.

A possible way around this would be to choose some initial values for the $2\ M\ c_l(\tilde{G}^i)$ and $c_r(\tilde{G}^i)$, and then compute the $2\ M\ \phi_l^i(\mathbf{x})$ and $\phi_r^i(\mathbf{x})$. A least-squares method could then be used to update the $2\ M\ c_l(\tilde{G}^i)$ and $c_r(\tilde{G}^i)$, after which the $2\ M\ \phi_l^i(\mathbf{x})$ and $\phi_r^i(\mathbf{x})$ would be recomputed. This iterative process would be continued until it converges in some sense. Whether or not such an iterative process is successful is an open question.

Of course, once this IT2 fuzzy system has been completely designed, then the $2\ M\ c_l(\tilde{G}^i)$ and $c_r(\tilde{G}^i)$ will be known so that the FBF matrix in (10.16) can be computed.

Some drawbacks to the least-squares method are: (1) how to choose the parameters of the antecedent FOUs is left as an open issue, and (2) how to choose the number of IT2 FBFs, M^*, is also left as an open issue. Usually, each variable is broken up into "enough" fuzzy sets so as to cover its domain interval with "enough" resolution. The centers of the Gaussian FOUs can be located at the centers of the intervals associated with each variable's fuzzy sets. The standard deviations can be chosen so that the FOUs have "sufficient" overlap.

The orthogonal least-squares procedure that was mentioned at the end of Sect. 4.2.2 as a way to select the most significant T1 FBFs has (to the best knowledge of this author) not been tried/extended as a way to select the most significant IT2 FBFs.

10.2.3 Derivative-Based Methods

In the derivative-based methods, none of the antecedent or consequent parameters of the IT2 fuzzy system are fixed ahead of time. They are all tuned using derivative-based optimization algorithms, such as steepest descent or Marquardt–Levenberg.

Given input–output training pairs $\{(\mathbf{x}^{(t)} : y^{(t)})\}_{t=1}^{N}$, the goal is to design an IT2 fuzzy system such that the following error function is minimized $(t = 1, \ldots, N)$:

$$J(\boldsymbol{\theta}) = e^{(t)} = \frac{1}{2}[y(\mathbf{x}^{(t)}) - y^{(t)}]^2 \tag{10.17}$$

in which $\boldsymbol{\theta}$ contains all of the design parameters, and $y(\mathbf{x})$ depends on the specific IT2 fuzzy system (e.g., see Examples 9.13–9.16). Clearly, $\partial J(\theta_j)/\partial\theta_j$ depends on $\partial y(\mathbf{x})/\partial\theta_j$.

Example 10.8 Some formulas for $\partial y(\mathbf{x})/\partial\theta_j$ are given in this example for the IT2 fuzzy systems in Examples 9.13–9.16, again beginning with Example 9.16 and working back to Example 9.13. These formulas are for when each θ_j is in only one IT2 FBF. The formulas for $\partial y(\mathbf{x})/\partial\theta_j$ when θ_j is not in an IT2 FBF [i.e., the λ^s in (10.1)] are very simple and are left to the reader.

- **BMM direct defuzzification fuzzy system**: From (9.183)–(9.185), it follows that (i = 1,…, M):

$$\partial y_{\mathrm{BMM}}(\mathbf{x})/\partial\theta_j = \sum_{i=1}^{M} y_i \partial\phi_l^i(\mathbf{x})/\partial\theta_j + \sum_{i=1}^{M} y_i \partial\phi_r^i(\mathbf{x})/\partial\theta_j \tag{10.18}$$

$$\frac{\partial\phi_l^i(\mathbf{x})}{\partial\theta_j} = m\frac{[\partial\underline{f}^i(\mathbf{x})/\partial\theta_j]\times\sum_{i=1}^{M}\underline{f}^i(\mathbf{x}) - \underline{f}^i(\mathbf{x})\times\sum_{i=1}^{M}\partial\underline{f}^i(\mathbf{x})/\partial\theta_j}{\left[\sum_{i=1}^{M}\underline{f}^i(\mathbf{x})\right]^2} \tag{10.19}$$

$$\frac{\partial\phi_r^i(\mathbf{x})}{\partial\theta_j} = n\frac{[\partial\bar{f}^i(\mathbf{x})/\partial\theta_j]\times\sum_{i=1}^{M}\bar{f}^i(\mathbf{x}) - \bar{f}^i(\mathbf{x})\times\sum_{i=1}^{M}\partial\bar{f}^i(\mathbf{x})/\partial\theta_j}{\left[\sum_{i=1}^{M}\bar{f}^i(\mathbf{x})\right]^2} \tag{10.20}$$

The derivatives $\partial\underline{f}^i(\mathbf{x})/\partial\theta_j$ and $\partial\bar{f}^i(\mathbf{x})/\partial\theta_j$ can be computed by using the appropriate formula for $\underline{f}^i(\mathbf{x})$ and $\bar{f}^i(\mathbf{x})$ in Table 10.4. It is at this point in the computations that one must know exactly where θ_j occurs. If it occurs in $\underline{\mu}_{F_i^s}(x_i)$, then $\partial\underline{\mu}_{F_i^s}(x_i)/\partial\theta_j$ will have to be computed, or if it occurs in $\bar{\mu}_{F_i^s}(x_i)$, then $\partial\bar{\mu}_{F_i^s}(x_i)/\partial\theta_j$ will have to be computed, etc. Examples of these calculations are given in Examples 6.16 (Gaussian primary MF with uncertain standard deviation) and 6.17 (Gaussian primary MF with uncertain mean).

- **NT direct defuzzification fuzzy system**: From (9.180) and (9.181), it follows that (l = 1,…, M):

$$\partial y_{NT}(\mathbf{x})/\partial\theta_j = \sum_{l=1}^{M} \bar{y}^l \partial\phi_l(\mathbf{x})/\partial\theta_j \tag{10.21}$$

$$\begin{aligned}\frac{\partial\phi_l(\mathbf{x})}{\partial\theta_j} &= \frac{\left[\partial\underline{f}^l(\mathbf{x})/\partial\theta_j + \partial\bar{f}^l(\mathbf{x})/\partial\theta_j\right]\times\left(\sum_{l=1}^{M}\left[\underline{f}^l(\mathbf{x})+\bar{f}^l(\mathbf{x})\right]\right)}{\left(\sum_{l=1}^{M}\left[\underline{f}^l(\mathbf{x})+\bar{f}^l(\mathbf{x})\right]\right)^2}\\ &\quad - \frac{\left[\underline{f}^l(\mathbf{x})+\bar{f}^l(\mathbf{x})\right]\times\left(\sum_{l=1}^{M}\left[\partial\underline{f}^l(\mathbf{x})/\partial\theta_j+\partial\bar{f}^l(\mathbf{x})/\partial\theta_j\right]\right)}{\left(\sum_{l=1}^{M}\left[\underline{f}^l(\mathbf{x})+\bar{f}^l(\mathbf{x})\right]\right)^2}\end{aligned} \tag{10.22}$$

The derivatives $\partial\underline{f}^l(\mathbf{x})/\partial\theta_j$ and $\partial\bar{f}^l(\mathbf{x})/\partial\theta_j$ can be computed as just described for the BMM direct defuzzification fuzzy system.

- **A2-C0 unnormalized IT2 TSK fuzzy system**: From (9.178) and (9.179), it follows that (l = 1,…, M):

$$\partial y_{TSK}^{U}(\mathbf{x})/\partial\theta_j = \sum_{l=1}^{M}\sum_{k=0}^{p} c_k^l \partial\phi_k^l(\mathbf{x})/\partial\theta_j \tag{10.23}$$

$$\frac{\partial\phi_k^l(\mathbf{x})}{\partial\theta_j} = \left[\frac{\partial\underline{f}^l(\mathbf{x})/\partial\theta_j + \partial\bar{f}^l(\mathbf{x})/\partial\theta_j}{2}\right] \times x_k \tag{10.24}$$

As above, the derivatives $\partial\underline{f}^l(\mathbf{x})/\partial\theta_j$ and $\partial\bar{f}^l(\mathbf{x})/\partial\theta_j$ can be computed as just described for the BMM direct defuzzification fuzzy system.

- **IT2 Mamdani fuzzy system with COS type-reduction + defuzzification**: From (9.172)–(9.174), it follows that[2] ($i = 1,\ldots, M$):

$$\partial y_{\mathrm{COS}}(\mathbf{x})/\partial\theta_j = \frac{1}{2}\sum_{i=1}^{M} c_l(\tilde{G}^i)\partial\phi_l^i(\mathbf{x})/\partial\theta_j + \frac{1}{2}\sum_{i=1}^{M} c_r(\tilde{G}^i)\partial\phi_r^i(\mathbf{x})/\partial\theta_j \tag{10.25}$$

$$\begin{aligned}\frac{\partial\phi_l^i(\mathbf{x})}{\partial\theta_j} = {} & \frac{\left[\delta_l^i\partial\bar{f}^i(\mathbf{x})/\partial\theta_j + (1-\delta_l^i)\partial\underline{f}^i(\mathbf{x})/\partial\theta_j\right] \times \sum_{i=1}^{M}\left[\delta_l^i\bar{f}^i(\mathbf{x}) + (1-\delta_l^i)\underline{f}^i(\mathbf{x})\right]}{\left(\sum_{i=1}^{M}\left[\delta_l^i\bar{f}^i(\mathbf{x}) + (1-\delta_l^i)\underline{f}^i(\mathbf{x})\right]\right)^2} \\ & - \frac{\left[\delta_l^i\bar{f}^i(\mathbf{x}) + (1-\delta_l^i)\underline{f}^i(\mathbf{x})\right] \times \sum_{i=1}^{M}\left[\delta_l^i\partial\bar{f}^i(\mathbf{x})/\partial\theta_j + (1-\delta_l^i)\partial\underline{f}^i(\mathbf{x})/\partial\theta_j\right]}{\left(\sum_{i=1}^{M}\left[\delta_l^i\bar{f}^i(\mathbf{x}) + (1-\delta_l^i)\underline{f}^i(\mathbf{x})\right]\right)^2}\end{aligned} \tag{10.26}$$

$$\begin{aligned}\frac{\partial\phi_r^i(\mathbf{x})}{\partial\theta_j} = {} & \frac{\left[\delta_r^i\partial\bar{f}^i(\mathbf{x})/\partial\theta_j + (1-\delta_r^i)\partial\underline{f}^i(\mathbf{x})/\partial\theta_j\right] \times \sum_{i=1}^{M}\left[\delta_r^i\bar{f}^i(\mathbf{x}) + (1-\delta_r^i)\underline{f}^i(\mathbf{x})\right]}{\left(\sum_{i=1}^{M}\left[\delta_r^i\bar{f}^i(\mathbf{x}) + (1-\delta_r^i)\underline{f}^i(\mathbf{x})\right]\right)^2} \\ & - \frac{\left[\delta_r^i\bar{f}^i(\mathbf{x}) + (1-\delta_r^i)\underline{f}^i(\mathbf{x})\right] \times \sum_{i=1}^{M}\left[\delta_r^i\partial\bar{f}^i(\mathbf{x})/\partial\theta_j + (1-\delta_r^i)\partial\underline{f}^i(\mathbf{x})/\partial\theta_j\right]}{\left(\sum_{i=1}^{M}\left[\delta_r^i\bar{f}^i(\mathbf{x}) + (1-\delta_r^i)\underline{f}^i(\mathbf{x})\right]\right)^2}\end{aligned} \tag{10.27}$$

It should be very clear from the formulas that are in this example that computing derivatives in an IT2 fuzzy system is complicated and it is very easy to make a mistake either in their computation or in the programming of the complicated

[2]Mendel (2004) is all about computing derivatives for IT2 Mamdani fuzzy system with COS type-reduction + defuzzification. Because it does not use the formulation of $y(\mathbf{x})$ that is given in (9.172)–(9.176), but instead uses the formulation of $y(\mathbf{x})$ that is in terms of the switch points $L(\mathbf{x})$ and $R(\mathbf{x})$, it is very complicated. This is due to having to reorder the IT2 FBFs, something that is not needed when (9.172)–(9.176) are used.

derivative formulas. Regardless, if one insists on using a derivative-based optimization algorithm, then the following procedure can be used for a specific IT2 fuzzy system and steepest descent:

1. Initialize all of the parameters in the antecedent and consequent FOUs.
2. Set the counter, e, of the training epoch to zero; i.e., $e \equiv 0$.
3. Set the counter, t, of the training data to unity; i.e., $t \equiv 1$.
4. Apply $p \times 1$ input $\mathbf{x}^{(t)}$ to the IT2 fuzzy system and compute the total firing interval for each rule; i.e., compute $\underline{f}^i$ and $\bar{f}^i$ $(i = 1, \ldots, p)$ using the formulas in the appropriate row of Table 10.4.
5. Compute $y(\mathbf{x}^{(t)})$ for the chosen IT2 fuzzy system.
6. Evaluate all of the partial derivatives (this assumes that you have programmed all of the detailed formulas for them) when $\mathbf{x} = \mathbf{x}^{(t)}$.
7. Update all of the design parameters using the following steepest descent algorithm:

$$\theta_j(t+1) = \theta_j(t) - \beta_{\theta_j} \frac{\partial J(\theta)}{\partial \theta_j} = \theta_j(t) - \beta_{\theta_j} [y(\mathbf{x}^{(t)}) - y^{(t)}] \frac{\partial}{\partial \theta_j} y(\mathbf{x}^{(t)}) \quad (10.28)$$

8. Set $t \equiv t+1$. If $t = N+1$, go to Step 9; otherwise, go to Step 4.
9. Set $e = e+1$. If $e = E$, STOP; otherwise, go to Step 3.

The discussions that are given in Sect. 4.2.3, following Example 4.7, about initializing (10.28), choosing the learning parameter β_{θ_j}, training for more than one epoch, and using a squared error function that depends on all N training data instead of (10.17) apply here as well.

All of the above discussions have been for an IT2 fuzzy system that can be expressed as an IT2 FBF expansion. Exercises 10.6–10.9 ask the reader to compute derivatives for some other IT2 fuzzy systems that can also be expressed as IT2 FBF expansions. An IT2 Mamdani fuzzy system that uses centroid type-reduction + defuzzification cannot be expressed as an IT2 FBF expansion. Computing derivatives for this fuzzy system is also left as an exercise for the reader (Exercise 10.11).

It is worth repeating that some drawbacks to derivative-based methods, in addition to the complexity of the derivative formulas, are:

1. If the UMF or LMF changes its mathematical formula over its domain, then derivative formulas change as well, and time-consuming domain tests have to be included.
2. They tend to only find a local extremum of an objective function rather than the global extremum, i.e., they tend to get trapped at a local extremum. Of course, there are ways to avoid getting trapped, but when derivatives are used there is a tendency to get trapped.
3. How to choose the number of FBFs, M^*, is left as an open issue. The next method can be used to resolve this drawback.

10.2.4 SVD-QR Method[3]

Recall, from Sect. 4.2.4 that the SVD–QR method is one that leads to rule-reduction. Its starting point for a T1 fuzzy system is the T1 FBF matrix (4.15). Rule reduction may also be needed for an IT2 fuzzy system. Modified versions of the Sect. 4.2.4 SVD–QR method, as described next, should let one accomplish this (Liang and Mendel 2000c; see also, Zhou et al. 2009).

The key is to begin with each of the IT2 FBF expansions that are given in Examples 9.13–9.16 after all parameters have been tuned, and proceed as indicated in the following example.

Example 10.9 Although this example is for the four IT2 fuzzy systems that have been focused on in earlier examples above, the approaches that are described next can also be applied to other IT2 fuzzy systems.

- **BMM direct defuzzification fuzzy system**: Observe that (9.183) has two kinds of IT2 FBFs, M of which are associated with just the lower firing levels and M of which are associated with just the upper firing levels. The following is a *five-step SVD-QR method*:

  1. Create an IT2 FBF matrix for the M ϕ_l^i in (9.184), called Φ_l, where

$$\Phi_l = \begin{pmatrix} \phi_l^1(\mathbf{x}^{(1)}) & \cdots & \phi_l^M(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ \phi_l^1(\mathbf{x}^{(N)}) & \cdots & \phi_l^M(\mathbf{x}^{(N)}) \end{pmatrix} \tag{10.29}$$

     Apply the first four steps of the Sect. 4.2.4 SVD-QR algorithm to Φ_l. This leads to $\hat{r}_l$ rules with rule numbers $J = \{j_1, j_2, \ldots, j_{\hat{r}_l}\}$, where $j_1, j_2, \ldots, j_{\hat{r}_l}$ are the rule numbers of the original M rules.

  2. Create an IT2 FBF matrix for the M ϕ_r^i in (9.185), called Φ_r, where

$$\Phi_r = \begin{pmatrix} \phi_r^1(\mathbf{x}^{(1)}) & \cdots & \phi_r^M(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ \phi_r^1(\mathbf{x}^{(N)}) & \cdots & \phi_r^M(\mathbf{x}^{(N)}) \end{pmatrix} \tag{10.30}$$

     Apply the first four steps of the Sect. 4.2.4 SVD-QR algorithm to Φ_r. This leads to $\hat{r}_r$ rules with rule numbers $K = \{k_1, k_2, \ldots, k_{\hat{r}_r}\}$, where $k_1, k_2, \ldots, k_{\hat{r}_r}$ are the rule numbers of the original M rules.

[3]This is a somewhat speculative section because only the SVD-QR method that is described for the IT2 Mamdani fuzzy system with COS type-reduction + defuzzification has been successfully applied in Liang and Mendel (2000c), although Example 10.9's Steps 4 and 5 are not explicitly mentioned in that paper, even though they were used.

3. Combine the results from Steps 1 and 2, i.e., keep the rules in $J \cup K$, which is the union of the rule-number sets J and K. Note that there will be some common rule numbers in J and K, so the total number of reduced rules $\hat{r}$ satisfies the following inequalities:

$$\max(\hat{r}_l, \hat{r}_r) \leq \hat{r} \leq (\hat{r}_l + \hat{r}_r) \tag{10.31}$$

   The $\hat{r}$ rules are renumbered $1, 2, \ldots, \hat{r}$, and there will now be $\hat{r} \equiv M'$ IT2 FBFs.
4. Renormalize the denominators of the $\hat{r}$ IT2 FBFs using the firing intervals for just those FBFs. The denominator of the IT2 FBFs whose numerator is a lower (upper) firing level should be the sum of all of the lower (upper) firing levels that are in all the IT2 FBFs with a lower (upper) firing level in their numerators. If this step is not performed, then the IT2 FBFs will still be normalized by the firing intervals of the original M IT2 FBFs, which defeats the purpose of the SVD-QR method.
5. Recompute the y_i coefficients using least squares. They are the λ^s in (10.1).

- **NT direct defuzzification fuzzy system**: Observe that (9.180) only has one kind of IT2 FBF, each of which involves the average of the firing interval for one of the M rules. The Sect. 4.2.4 SVD-QR algorithm can be directly applied directly to the following IT2 FBF matrix:

$$\Phi_{NT} = \begin{pmatrix} \phi_1(\mathbf{x}^{(1)}) & \cdots & \phi_M(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}^{(N)}) & \cdots & \phi_M(\mathbf{x}^{(N)}) \end{pmatrix} \tag{10.32}$$

  The entries in (10.32) are computed by using (9.181).

- **A2-C0 unnormalized IT2 TSK fuzzy system**: Observe that (9.178) also has only one kind of IT2 FBF, each of which also involves the average of the firing interval for one of the M rules; however, those averages are also multiplied by an x_j. There are $M(p+1)$ IT2 FBFs. The Sect. 4.2.4 SVD-QR algorithm can be applied directly to the following IT2 FBF matrix:

$$\Phi_{A2-C0} = \begin{pmatrix} \phi_0^1(\mathbf{x}^{(1)}) & \cdots & \phi_0^M(\mathbf{x}^{(1)}) & \phi_1^1(\mathbf{x}^{(1)}) & \cdots & \phi_1^M(\mathbf{x}^{(1)}) & \cdots & \phi_p^1(\mathbf{x}^{(1)}) & \cdots & \phi_p^M(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \phi_0^1(\mathbf{x}^{(N)}) & \cdots & \phi_0^M(\mathbf{x}^{(N)}) & \phi_1^1(\mathbf{x}^{(N)}) & \cdots & \phi_1^M(\mathbf{x}^{(N)}) & \cdots & \phi_p^1(\mathbf{x}^{(N)}) & \cdots & \phi_p^M(\mathbf{x}^{(N)}) \end{pmatrix} \tag{10.33}$$

  The entries in (10.33) are computed by using (9.179).

- **IT2 Mamdani fuzzy system with COS type-reduction + defuzzification**: Observe that (9.172) has two kinds of IT2 FBFs that are given in (9.173) and (9.174), respectively. All of the 2 M IT2 FBFs are associated with different combinations of lower and upper firing levels. The five-step SVD-QR procedure given above for the BMM direct defuzzification fuzzy system can be applied here as well, but now the entries in (10.29) are computed using (9.173), whereas the entries in (10.30) are computed using (9.174).

Although the above SVD–QR design method provides optimal values for λ^s in (10.1) in a least-squares sense, it does not provide any values for the remaining design parameters of the fuzzy system. The antecedent FOU parameters must be pre-specified to use the SVD-QR design method. See, also Sect. 10.2.6.

10.2.5 Derivative-Free Methods

Paraphrasing the first paragraph of Sect. 4.2.5: There are now many optimization methods that do not require derivatives of mathematical objective functions. They are very attractive for tuning FOU parameters in an IT2 fuzzy system, because they are a way to overcome the first two drawbacks to derivative-based methods that are listed at the end of Sect. 10.2.3, namely, having to use domain tests so that a correct derivative formula is used, and getting trapped at a local extremum.

The discussions about derivative-free methods (algorithms) that are given in Sect. 4.2.5 apply as well for IT2 fuzzy systems, with some modifications that are explained below.

Focusing on QPSO (as was done for T1 fuzzy systems), the following four-step design procedure is advocated when QPSO (or any particle-based method) is used to optimize the parameters of an IT2 fuzzy system:

1. Design a T1 fuzzy system using QPSO. If it is a singleton T1 fuzzy system, then initialize all of its particles randomly. If it is a non-singleton T1 fuzzy system, then follow the two-step procedure that is described in Sect. 4.2.5. If system performance is acceptable, stop; otherwise proceed to Step 2.
2. Design a singleton IT2 fuzzy system by optimizing its parameters using QPSO in which one particle is associated with the just designed T1 fuzzy system, and where the parameters of all the remaining particles are initialized randomly. If system performance is acceptable, stop; otherwise proceed to Step 3.
3. Design a T1 non-singleton IT2 fuzzy system by optimizing its parameters using QPSO in which one particle is associated with the just designed singleton IT2 fuzzy system, and where the parameters of all the remaining particles are initialized randomly. If system performance is acceptable, stop; otherwise proceed to Step 4.
4. Design an IT2 non-singleton IT2 fuzzy system by optimizing its parameters using QPSO in which one particle is associated with the just designed T1 non-singleton IT2 fuzzy system, and where the parameters of all the remaining

Table 10.5 Pseudo-Code for QPSO as used in optimal designs of IT2 fuzzy systems Mendel (2014)

Initialize $\boldsymbol{\theta}_1(1)$ as a previously designed fuzzy system[a] particle, and all other $\boldsymbol{\theta}_m(1)$ randomly $(m = 2, \ldots, N_m)$
Set $\mathbf{p}_m(1) = \boldsymbol{\theta}_m(1)$ $(m = 1, \ldots, N_m)$
For t = 1 to $G - 1$
Calculate $\mathbf{m}(t) = \frac{1}{N_m}\sum_{m=1}^{N_m}\mathbf{p}_m(t)$
Calculate $J(\mathbf{p}_m(t))$ $(m = 1, \ldots, N_m)$
Calculate $\mathbf{p}_{gbest}(t) = arg \min_{\mathbf{p}_m(t), \forall m=1,\ldots,N_m} J(\mathbf{p}_m(t))$
for m = 1 to N_m(number of particles)
Calculate $J(\boldsymbol{\theta}_m(t))$
If $J(\boldsymbol{\theta}_m(t)) < J(\mathbf{p}_m(t))$ $\mathbf{p}_m(t) = \boldsymbol{\theta}_m(t)$ **end if**
for j = 1 to N_θ (number of components in each particle)

$\eta = rand(0,1)$
$p_{m,j}(t+1) = \eta \times p_{m,j}(t) + (1-\eta) \times p_{gbest,j}(t)$
$\rho = rand(0,1)$
if $rand(0,1) > 0.5$ **then**
$\theta_{m,j}(t+1) = p_{m,j}(t+1) - \beta\lvert m_j(t) - \theta_{m,j}(t)\rvert \ln(1/\rho)$
else
$\theta_{m,j}(t+1) = p_{m,j}(t+1) + \beta\lvert m_j(t) - \theta_{m,j}(t)\rvert \ln(1/\rho)$
end if
end for
end for

end for

[a]This could be a non-singleton T1, singleton IT2, or T1 non-singleton IT2 design

particles are initialized randomly. If system performance is acceptable, stop; otherwise design a GT2 fuzzy system (see Chap. 11, especially Example 11.10).

Pseudo-code for QPSO that abides by this two-step procedure is given in Table 10.5.

Theorem 10.1 *By virtue of the QPSO algorithm, the performance of the optimized singleton IT2 fuzzy system cannot be worse than that of the optimized non-singleton type-1 fuzzy system, the performance of the optimized T1 non-singleton IT2 fuzzy system cannot be worse than that of the optimized singleton IT2 fuzzy system, and the performance of the optimized IT2 non-singleton IT2 fuzzy system cannot be worse than that of the optimized T1 non-singleton IT2 fuzzy system.*

Proof Because the proof of this theorem is so similar to the proof of Theorem 4.1, it is left as some exercises for the reader. See Exercises 10.12–10.14.

By this four-step systematic design approach it is not possible for the performance of an optimized singleton IT2 fuzzy system to be worse than that of an optimized non-singleton T1 fuzzy system,[4] and the performance of an optimized T1 non-singleton IT2 fuzzy system to be worse than that of an optimized singleton IT2 fuzzy system, and the performance of an optimized IT2 non-singleton IT2 fuzzy system to be worse than that of an optimized T1 non-singleton IT2 fuzzy system. This does not mean that the performance of the optimized singleton IT2 fuzzy system will be *significantly better* than that of the optimized non-singleton T1 fuzzy system, or that the performance of the optimized T1 non-singleton IT2 fuzzy system will be *significantly better* than that of the optimized singleton IT2 fuzzy system, or that performance of the optimized IT2 non-singleton IT2 fuzzy system will be *significantly better* than that of the optimized T1 non-singleton IT2 fuzzy system. There is no analysis that is available to-date that focuses on such relative performance improvements. Of course, relative improvements are very application dependent because objective function $J(\boldsymbol{\theta})$ is application dependent.

Example 10.10 The trick to setting up an IT2 fuzzy system particle is to do it first for the most complicated IT2 fuzzy system, and to then show how a particle for a less complicated IT2 fuzzy system can be embedded in it. This is done here for an *IT2 non-singleton IT2 Mamdani fuzzy system with COS type-reduction + defuzzification*. Exercise 10.15 asks the reader to repeat this example for other IT2 fuzzy systems. Recall that this chapter assumes that all FOUs are for Gaussian primary MFs with uncertain standard deviations, so they are described by three design parameters $\{m, \sigma_1, \sigma_2\}$.

The structure of a particle for an IT2 non-singleton IT2 fuzzy system is:

$$\boldsymbol{\phi}_{\text{IT2}}^{\text{IT2NS}} = \text{col}\Bigg(\underbrace{\overbrace{m_1^1, \sigma_{1,1}^1, \sigma_{1,2}^1}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^1, \sigma_{p,1}^1, \sigma_{p,2}^1}^{\text{Antecedent } p}, \overbrace{c_l(\tilde{G}^1), c_r(\tilde{G}^1)}^{\text{Consequent}}}_{\text{Rule 1}}; \ldots; \underbrace{\overbrace{m_1^M, \sigma_{1,1}^M, \sigma_{1,2}^M}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^M, \sigma_{p,1}^M, \sigma_{p,2}^M}^{\text{Antecedent } p}, \overbrace{c_l(\tilde{G}^M), c_r(\tilde{G}^M)}^{\text{Consequent}}}_{\text{Rule M}}; \underbrace{\overbrace{\sigma_{1,1}, \sigma_{1,2}}^{\text{Input 1}}, \ldots, \overbrace{\sigma_{p,1}, \sigma_{p,2}}^{\text{Input } p}}_{\text{IT2 non-singleton fuzzifier}}\Bigg) \tag{10.34}$$

Each rule has p antecedents that are described by three parameters and a consequent that is described by the two endpoints of its centroid. Each input is centered about its measured value and its FOU is described by two parameters.

[4]If an IT2 fuzzy system cannot be designed by means of optimization, but can only be designed by trial and error, then it is possible that the performance of the singleton IT2 fuzzy system may be worse than that of a non-singleton type-1 fuzzy system, or the performance of the T1 non-singleton IT2 fuzzy system may be worse than that of a singleton IT2 fuzzy system, or the performance of the IT2 non-singleton IT2 fuzzy system may be worse than that of a T1 non-singleton IT2 fuzzy system. This occurs because it may not be possible to try all possible combinations of the design parameters, and is therefore the result of an incomplete design procedure. One must be very cautious about drawing conclusions from such trial and error designs.

The T1 non-singleton IT2 fuzzy system particle, which must be of the same length as this IT2 non-singleton IT2 fuzzy system particle for it to be embedded in such a particle, begins with (10.34) and expresses it as:

$$\boldsymbol{\phi}_{\mathrm{IT2}}^{\mathrm{T1NS}} = \mathrm{col}\left(\underbrace{\overbrace{m_1^1, \sigma_{1,1}^1, \sigma_{1,2}^1}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^1, \sigma_{p,1}^1, \sigma_{p,2}^1}^{\text{Antecedent } p}, \overbrace{c_l(\tilde{G}^1), c_r(\tilde{G}^1)}^{\text{Consequent}};}_{\text{Rule1}} \ldots; \underbrace{\overbrace{m_1^M, \sigma_{1,1}^M, \sigma_{1,2}^M}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^M, \sigma_{p,1}^M, \sigma_{p,2}^M}^{\text{Antecedent } p}, \overbrace{c_l(\tilde{G}^M), c_r(\tilde{G}^M)}^{\text{Consequent}};}_{\text{RuleM}} \underbrace{\overbrace{\sigma_1, \sigma_1}^{\text{Input 1}}, \ldots, \overbrace{\sigma_p, \sigma_p}^{\text{Input } p}}_{\text{T1 non-singleton fuzzifier}} \right) \tag{10.35}$$

Observe, in (10.35), that: (1) All of the FOU parameters are taken from the QPSO-optimized T1 non-singleton IT2 fuzzy system design; (2) By setting the two values for all of the fuzzifier standard deviations equal to the same σ, the IT2 non-singleton fuzzifier reduces to a T1 non-singleton fuzzifier; and, (3) In this way, it is straightforward to embed a T1 non-singleton IT2 fuzzy system particle into an IT2 non-singleton IT2 fuzzy system particle.

The singleton IT2 fuzzy system particle, which must be of the same length as this T1 non-singleton IT2 fuzzy system particle for it to be embedded in such a particle, begins with (10.35) and expresses it as:

$$\boldsymbol{\phi}_{\mathrm{IT2}}^{\mathrm{S}} = \mathrm{col}\left(\underbrace{\overbrace{m_1^1, \sigma_{1,1}^1, \sigma_{1,2}^1}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^1, \sigma_{p,1}^1, \sigma_{p,2}^1}^{\text{Antecedent } p}, \overbrace{c_l(\tilde{G}^1), c_r(\tilde{G}^1)}^{\text{Consequent}};}_{\text{Rule 1}} \ldots; \underbrace{\overbrace{m_1^M, \sigma_{1,1}^M, \sigma_{1,2}^M}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^M, \sigma_{p,1}^M, \sigma_{p,2}^M}^{\text{Antecedent } p}, \overbrace{c_l(\tilde{G}^M), c_r(\tilde{G}^M)}^{\text{Consequent}};}_{\text{Rule M}} \underbrace{\overbrace{0, 0}^{\text{Input 1}}, \ldots, \overbrace{0, 0}^{\text{Input } p}}_{\text{singleton fuzzifier}} \right) \tag{10.36}$$

Observe in (10.36), that: (1) All of the FOU parameters are taken from the QPSO-optimized singleton IT2 fuzzy system design; (2) By setting the two values for all of the fuzzifier standard deviations equal to 0, the T1 non-singleton fuzzifier reduces to a singleton fuzzifier; and, (3) In this way, it is straightforward to embed a singleton IT2 fuzzy system particle into a T1 non-singleton IT2 fuzzy system particle.

The non-singleton T1 fuzzy system particle, which must be of the same length as this singleton IT2 fuzzy system particle for it to be embedded in such a particle, begins with (10.36) and expresses it as:

$$
\begin{aligned}
\boldsymbol{\phi}_{\mathrm{T1}}^{\mathrm{NS}} = \mathrm{col}\Bigg(& \underbrace{\overbrace{m_1^1, \sigma_1^1, \sigma_1^1}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^1, \sigma_p^1, \sigma_p^1}^{\text{Antecedent } p}, \overbrace{y_{G^1}, y_{G^1}}^{\text{Consequent}}}_{\text{Rule 1}}; \ldots; \underbrace{\overbrace{m_1^M, \sigma_1^M, \sigma_1^M}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^M, \sigma_p^M, \sigma_p^M}^{\text{Antecedent } p}, \overbrace{y_{G^M}, y_{G^M}}^{\text{Consequent}}}_{\text{Rule M}}; \\
& \underbrace{\overbrace{\sigma_1, \sigma_1}^{\text{Input 1}}, \ldots, \overbrace{\sigma_p, \sigma_p}^{\text{Input } p}}_{\text{non-singleton fuzzifier}} \Bigg)
\end{aligned}
\tag{10.37}
$$

Observe, in (10.37), that: (1) All of the MF parameters are taken from the QPSO-optimized non-singleton T1 fuzzy system design; (2) By setting the values for the endpoints of the standard deviations for an antecedent FOU to be the same, and the endpoints of the centroids for a consequent to be the same, the uncertainties in each rule about the standard deviations for all p antecedents, and its consequent, have disappeared; (3) By setting the two values for all of the fuzzifier standard deviations equal to the same σ, the singleton fuzzifier in (10.36) expands to a non-singleton fuzzifier; and, (4) In this way, it is straightforward to embed a non-singleton T1 fuzzy system particle into a singleton IT2 fuzzy system particle.

Finally, the singleton T1 fuzzy system particle, which must be of the same length as this non-singleton T1 fuzzy system particle for it to be embedded in such a particle, begins with (10.37) and expresses it as:

$$
\begin{aligned}
\boldsymbol{\phi}_{\mathrm{T1}}^{\mathrm{S}} = \mathrm{col}\Bigg(& \underbrace{\overbrace{m_1^1, \sigma_1^1, \sigma_1^1}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^1, \sigma_p^1, \sigma_p^1}^{\text{Antecedent } p}, \overbrace{y_{G^1}, y_{G^1}}^{\text{Consequent}}}_{\text{Rule 1}}; \ldots; \underbrace{\overbrace{m_1^M, \sigma_1^M, \sigma_1^M}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^M, \sigma_p^M, \sigma_p^M}^{\text{Antecedent } p}, \overbrace{y_{G^M}, y_{G^M}}^{\text{Consequent}}}_{\text{Rule M}}; \\
& \underbrace{\overbrace{0, 0}^{\text{Input 1}}, \ldots, \overbrace{0, 0}^{\text{Input } p}}_{\text{singleton fuzzifier}} \Bigg)
\end{aligned}
\tag{10.38}
$$

Observe, in (10.38), that: (1) All of the MF parameters are taken from the QPSO-optimized singleton T1 fuzzy system design; (2) By setting the two values for all of the fuzzifier standard deviations equal to 0, the non-singleton fuzzifier reduces to a singleton fuzzifier; and, (3) In this way, it is straightforward to embed a singleton T1 fuzzy system particle into a non-singleton T1 fuzzy system particle.

10.2.6 Iterative Design Methods

There are many ways to combine the optimization methods that have been described in Sects. 10.2.1–10.2.5, such as:

- By combining the SVD–QR method with a derivative-based or particle-based method, one can design all of the parameters of an IT2 fuzzy system including the number of the most significant rules, M'. The following iterative design method can be very successful:

  1. Fix the number of rules, M, at a reasonable value.
  2. Use a derivative-based or particle-based method to design all the antecedent and consequent FOU parameters.
  3. Apply the SVD–QR method to the results of the derivative-based or particle-based method to determine $M' < M$ IT2 FBFs.
  4. Renormalize the IT2 FBFs and re-compute the linear combining parameters using least-squares.
  5. If performance is acceptable, STOP. Otherwise, return to Step 2 for a re-tuning of the antecedent and consequent parameters.

- By applying the SVD-QR method to the IT2 FBF matrix that can be created after IT2 Zadeh rules are obtained from the IT2 WM method.
- By using an evolutionary or bio-inspired optimization method that is set up not only to optimize FOU parameters, but also other things such as [e.g., Rutkowski (2004)]: which antecedents to use as well as their number (i.e., p), the number of linguistic terms for each variable (i.e., $Q_1, \ldots, Q_p$), the number of rules (i.e., M), the t-norm used (i.e., minimum or product), Mamdani product or minimum, and the type-reduction method (e.g., height or COS).

10.2.7 Remarks

10.2.7.1 General Remarks

The following objection to optimal IT2 fuzzy system designs is sometimes raised: Because an IT2 fuzzy system is described by more parameters than is a T1 fuzzy system, it is unfair to compare the performance from such an IT2 fuzzy system with the T1 fuzzy system, that is, it is only fair to compare optimal designs for IT2 and T1 fuzzy systems that have exactly the same number of parameters. Interestingly, a similar objection is not raised when optimal designs are compared for a T1 fuzzy system and a non-fuzzy system, in which the T1 fuzzy system has more design degrees of freedom than the non-fuzzy system. The design approach advocated in this book is one that first begins with a T1 fuzzy system and tries to achieve the desired performance. It is only when such desired performance cannot be met that this book advocates moving up to an IT2 fuzzy system.

It is worth restating some of the general remarks that are given in Sect. 4.2.7 but in the context of IT2 fuzzy system designs.

When an IT2 fuzzy system is going to be used as part of a consumer (or military) product then it should be designed to meet pre-specified performance specifications instead of by optimizing a mathematical objective function (unless, perhaps, the performance specifications can be expressed by means of such a function). If this can be accomplished by using a singleton IT2 fuzzy system, the design can stop, but if it cannot then the design process should proceed first to a T1 non-singleton IT2 fuzzy system and then to an IT2 non-singleton IT2 fuzzy system. Unfortunately, such designs do not appear in the open literature because usually product performance specifications are proprietary or classified. Consequently, academics (such as the author) focus on optimal designs.

Finally, there are additional benefits to using an IT2 fuzzy system over a T1 fuzzy system that may not be encapsulated by a mathematical objective function, or, to put it another way, a design that is based only on minimizing a mathematical objective function may not in the end be evaluated for its success just by examining the minimum (or maximum) value for that objective function. For example, even greater smoothness can be obtained by using IT2 fuzzy sets over T1 fuzzy sets, and smoothness may be difficult to quantify using an objective function.

10.2.7.2 The Marín-Valencia-Sáez (MVS) Design Procedure for IT2 Fuzzy Systems

Marín et al. (2016) is arguably the first article to make use of the type-reduced set in an active[5] way during the optimal design of an IT2 fuzzy system. Their approach is to obtain the parameters of a Mamdani IT2 fuzzy system that uses COS type-reduction + defuzzification by minimizing a new and very novel objective function. In order to explain this objective function, let:

1. *PICP* denote the *sample prediction interval coverage probability*, where:

$$PICP \equiv \frac{1}{N}\sum_{t=1}^{N} \delta_t \tag{10.39}$$

$$\delta_t = \begin{cases} 1 & \text{if } y^{(t)} \in [y_l^{\mathrm{COS}}(\mathbf{x}^{(t)}), y_r^{\mathrm{COS}}(\mathbf{x}^{(t)})] \\ 0 & \text{if } y^{(t)} \notin [y_l^{\mathrm{COS}}(\mathbf{x}^{(t)}), y_r^{\mathrm{COS}}(\mathbf{x}^{(t)})] \end{cases} \tag{10.40}$$

   $y^{(t)}$ is the tth measured value of a signal of interest, and is an element in the training sample. In their design procedure that is explained below, PICP is set equal to a desired level called $1-\alpha$ that has to specified by the designer (just as a confidence level has to be specified in statistics).

[5]In everything that has been described prior to this section the type-reduced set has only been a means to an end, the end being the defuzzified output. Although this author has also used the type-reduced set in this way, he always felt that it contained valuable information about dispersion about the output and that somehow this information ought to also be used in a design, since dispersion about the mean is used in probability based designs.

2. *PINAW* denotes the *prediction interval normalized average width*, where

$$PINAW = \frac{1}{N}\sum_{t=1}^{N}[y_r^{\mathrm{COS}}(\mathbf{x}^{(t)}) - y_l^{\mathrm{COS}}(\mathbf{x}^{(t)})] \tag{10.41}$$

   Since the length of the COS type-reduced set provides a measure of all MF uncertainties in the IT2 fuzzy system, one would like *PINAW* to be as small as possible. By trying to make *PINAW* as small as possible and *PICP* at least as large as $1-\alpha$, one is trying to force the output of the IT2 fuzzy system to be in as tight an uncertainty interval as possible, and as often as possible.
3. $||e||_2$ denotes the RMSE between the output of the IT2 fuzzy system and the data in the training set, where

$$||e||_2 = \sqrt{\frac{1}{N}\sum\nolimits_{t=1}^{N}[y(\mathbf{x}^{(t)}) - y_{IT2}(\mathbf{x}^{(t)})]^2} \tag{10.42}$$

$$y_{IT2}(\mathbf{x}^{(t)}) = \frac{1}{2}[y_r^{\mathrm{COS}}(\mathbf{x}^{(t)}) + y_l^{\mathrm{COS}}(\mathbf{x}^{(t)})] \tag{10.43}$$

   One would also like e (or e^2) to be as small as possible.

The initial statement of their design problem is: let $\boldsymbol{\theta}$ be a vector that contains all of the IT2 fuzzy system design parameters; then find $\boldsymbol{\theta}$ that minimizes $PINAW + ||e||_2^2$ such that $PICP = 1-\alpha$. Marín et al. (2016) state:

> The solution [to this optimization problem] provides the parameters $\boldsymbol{\theta}$ of the premises and consequences of each rule in such a way that the prediction error of the expected trajectory and the width of the resulting interval [type-reduced set] are minimized and such that the desired coverage probability is guaranteed.

Because $PICP = 1-\alpha$ is a hard constraint, they reformulate this optimization problem as the following unconstrained minimization problem: Minimize $J(\boldsymbol{\theta})$, where

$$J(\boldsymbol{\theta}) = \beta_1 PINAW + \beta_2||e||_2^2 + \exp[-\eta(PICP - (1-\alpha))] \tag{10.44}$$

In (10.44), $\exp[-\eta(PICP-(1-\alpha))]$ is called a *barrier function*. If $PICP \le 1-\alpha$, then the argument of the exponential becomes positive and so the barrier function will be large; on the other hand, if $PICP > 1-\alpha$, then the argument of the exponential becomes negative and so the barrier function will be small. In (10.44), the weights β_1, β_2 and η have to be specified a priori (or rules can be learned about them, so that these weights can be adapted during the training). Marín et al. (2016) state:

> In (10.44), the weighting factors must be chosen such that if $PICP \le 1-\alpha$, the term $\exp[-\eta(PICP-(1-\alpha))]$ is the dominant term in the cost function; otherwise, the *PINAW* and $||e||_2^2$ are the long-term influential factors.

This is a very different optimization problem from the one that almost everyone (maybe everyone) else has considered. All others have found $\boldsymbol{\theta}$ that minimizes $||e||_2^2$.

10.3 Case Study: Forecasting of Time-Series

This section first extends the time-series forecasting case study of Sect. 4.3, when the additive measurement noise is stationary, to singleton and T1 non-singleton IT2 fuzzy systems. It then examines a time-series forecasting problem in which the additive noise is nonstationary.

10.3.1 Forecasting of Time Series When the Measurement Noise Is Stationary

The problem of forecasting a time-series is described in Sect. 4.3. Here two IT2 fuzzy system forecasters (singleton and T1 non-singleton IT2 Mamdani fuzzy systems) are designed for the Mackey–Glass time-series using derivative-based (steepest descent) designs. Results from those designs are then compared with results obtained from the derivative-based (steepest descent) designs for singleton and non-singleton T1 fuzzy systems which are summarized in Fig. 4.7. The T1 non-singleton IT2 fuzzy system forecaster represents, for the first time, a fuzzy system that accounts for *all* of the uncertainties that are present, namely, rule uncertainties due to training with noisy data and measurement uncertainties due to noisy measurements that are used during actual forecasting.

Both designs were based on the following 1000 *noisy* data points: $x(1001), x(1002), \ldots, x(2000)$. The first 504 noisy data, $x(1001), x(1002), \ldots,$ $x(1504)$, were used for training, i.e., for the designs of the IT2 fuzzy system forecasters, whereas the remaining 496 noisy data, $x(1505), x(1506), \ldots, x(2000)$, were used for testing the designs. The noise-free Mackey–Glass time-series is depicted in Fig. 4.3. As in Sect. 4.3.4, the noise-free sampled time-series, $s(k)$, is corrupted by uniformly distributed stationary additive noise, $n(k)$, so that

$$x(k) = s(k) + n(k) \quad k = 1001, 1002, \ldots, 2000 \tag{10.45}$$

where $\mathrm{SNR} = 0\,\mathrm{dB}$. One realization of $x(1001), x(1002), \ldots, x(2000)$ is depicted in Fig. 4.4.

As in Sect. 4.3.6, rules for forecasting $x(k+1)$ had four antecedents, namely, $x(k-3)$, $x(k-2)$, $x(k-1)$, and $x(k)$, and two fuzzy sets were used for each antecedent, hence, a total of 16 rules were used. Gaussian primary MFs of uncertain means (Example 6.17) were chosen for the antecedents of both IT2 fuzzy systems, and each of their rules was characterized by 12 antecedent MF parameters (the left- and right-hand bounds on the mean, and the standard deviation for each of the four Gaussian MFs) and two consequent parameters (the left- and right-hand endpoints for the centroid of the consequent T2 FS). Product t-norm and COS type-reduction were used.

Both of IT2 fuzzy system designs used their respective *partially dependent approaches* that are described toward the end of Sect. 10.1.

10.3.1.1 Singleton IT2 Fuzzy System Forecaster

First, the best possible singleton and non-singleton T1 fuzzy systems were designed (as described in Sects. 4.3.3 and 4.3.6) and then some of their parameters were used to initialize the design of a singleton IT2 Mamdani fuzzy system.

The RMSE performances of the T1 fuzzy system designs were computed using (4.51) for the singleton T1 fuzzy system and (4.53) for the non-singleton T1 fuzzy system, and the RMSE performance of the singleton IT2 fuzzy system was computed, as:

$$\mathrm{RMSE}_{s2} = \sqrt{\frac{1}{496}\sum_{k=1504}^{1999}\left[s(k+1) - y_{s2}(\mathbf{x}^{(k)})\right]^2} \tag{10.46}$$

In (10.46), y_{s2} was computed using (9.113), (9.112) and (9.111), where the firing interval was computed using (9.25) and (9.26).

The intervals of uncertainty for the means of each of the four antecedent's two fuzzy sets were set initially to $[m_x - 2\sigma_x - 0.25\sigma_n, m_x - 2\sigma_x + 0.25\sigma_n]$ and $[m_x + 2\sigma_x - 0.25\sigma_n, m_x + 2\sigma_x + 0.25\sigma_n]$, respectively, where m_x and σ_x are the mean and standard deviation, respectively, of the data in the 504 training samples, $x(1001), x(1002), \ldots, x(1504)$. For the initial value of the standard deviation of the additive noise, σ_n, the final tuned result for the standard deviation of the input, σ_X, obtained from the non-singleton T1 fuzzy system design, was used. Additionally, $c_r(\tilde{G}^i)$ and $c_l(\tilde{G}^i)$ were initially chosen as $\bar{y}^i + \sigma_n$ and $\bar{y}^i - \sigma_n$, respectively, where $\bar{y}^i$ was also obtained from the T1 singleton fuzzy system design.

As stated earlier, the parameters that were tuned during the steepest descent design of the singleton IT2 Mamdani fuzzy system are the left- and right-hand bounds on the mean and the standard deviation for each of the four Gaussian primary MFs, and the left- and right-hand endpoints for the centroid of each consequent T2 fuzzy set. The total number of parameters tuned in this design is 14 parameters per rule × 16 rules = 224 parameters.

The singleton IT2 Mamdani fuzzy system was tuned using a steepest descent algorithm in which the learning parameter $\beta_\theta = 0.2$. Training and testing were carried out for six epochs. After each epoch, the testing data was used to see how this fuzzy system performed, by computing $\mathrm{RMSE}_{s2}(SD)$ using (10.46). This entire process was repeated 50 times using 50 independent sets of 1000 data points, at the

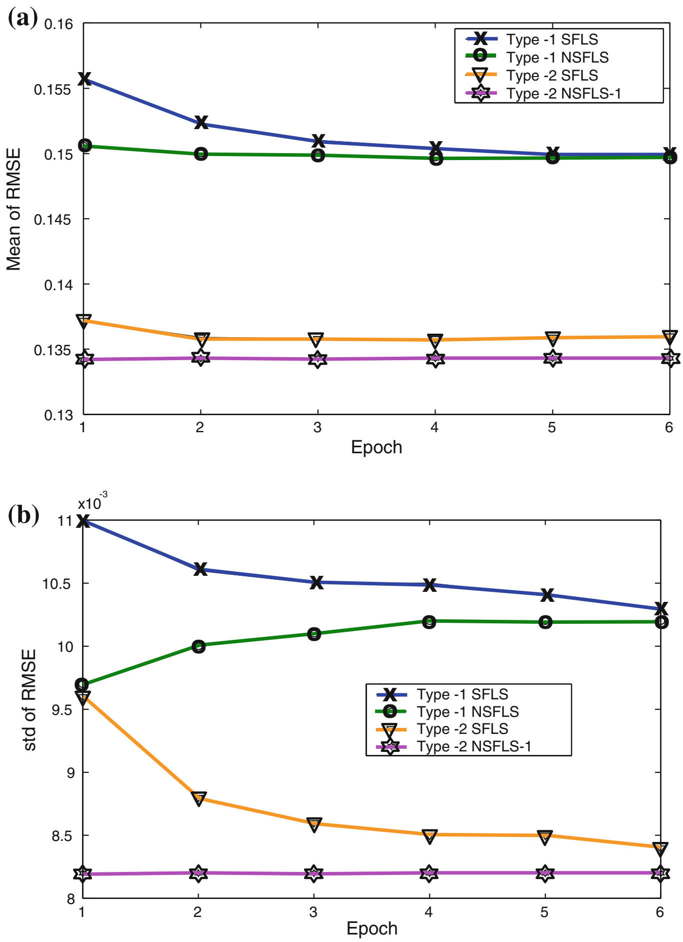


Fig. 10.1 The mean and standard deviation of $\text{RMSE}_s(\text{SD})$, $\text{RMSE}_{ns}(\text{SD})$, $\text{RMSE}_{s2}(\text{SD})$, and, $\text{RMSE}_{ns2-1}(\text{SD})$ averaged over 50 Monte Carlo designs. Tuning was performed in each realization for six epochs. **a** Mean values, and **b** standard deviation values. Note that "FLS" is equivalent to "fuzzy system"

end of which 50 $\text{RMSE}_{s2}(SD)$ values were obtained. The average values and standard deviations of $\text{RMSE}_{s2}(SD)$ as well $\text{RMSE}_{s}(SD)$ and $\text{RMSE}_{ns}(SD)$ [from Figs. 4.7a, b] are plotted in Fig. 10.1 for each of the six epochs. Observe that:

- The top two curves for the T1 fuzzy systems are the same as those in Fig. 4.7.
- There is a *substantial* improvement in performance (in both the mean and standard deviation of the RMSE) for the singleton IT2 fuzzy system over its type-1 counterparts.

The reason for this substantial improvement in performance is that for the first time the uncertainties that are in the training data have been incorporated into the rules of the fuzzy system forecaster. Although the reduction in the final average RMSE, from around 0.15 to 0.136, only represents a reduction of $9\frac{1}{3}\%$, if this was a financial time-series, a lot of money could be made with a $9\frac{1}{3}\%$ improvement in forecasting performance. The reduction in the standard deviation of the RMSE, from around 10.2×10^{-3} to 8.4×10^{-3}, represents a reduction of more than 17%.

Although the singleton IT2 fuzzy system forecaster has incorporated the uncertainties that are in the training data into its rules, it still does not account for the input measurement uncertainties because it is using singleton fuzzification. Next, a T1 non-singleton IT2 fuzzy system is designed that accounts for all of the uncertainties that are present in the forecasting problem.

10.3.1.2 T1 Non-singleton IT2 Fuzzy System Forecaster

For illustrative purposes, the partially dependent approach, that is, described in Sect. 10.1 for the tuning of a T1 non-singleton IT2 Mamdani fuzzy system, was used. More specifically, all of the parameters from the just designed singleton IT2 fuzzy system were kept, and only the new parameters of the T1 non-singleton IT2 Mamdani fuzzy system were tuned. In the present case, there is only one new parameter, namely, σ_X; i.e., the same standard deviation was used for each of the four input measurement MFs, because the inputs are delayed versions of one another.

The RMSE performance of the T1 non-singleton IT2 fuzzy system was computed, as:

$$\mathrm{RMSE}_{ns2-1} = \sqrt{\frac{1}{496}\sum_{k=1504}^{1999}\left[s(k+1) - y_{ns2-1}(\mathbf{x}^{(k)})\right]^2} \tag{10.47}$$

In (10.47), y_{ns2-1} was computed using (9.113), (9.112) and (9.111), where the firing interval was computed using (9.51) and (9.52).

The T1 singleton IT2 Mamdani fuzzy system was tuned only for σ_X using a steepest descent algorithm in which σ_X was initialized by using the final tuned result for it from the non-singleton T1 fuzzy system design, and the learning parameter $\beta_\theta = 0.2$. Training and testing were again carried out for six epochs. After each epoch, the testing data was used to see how this fuzzy system performed, by computing $\mathrm{RMSE}_{ns2-1}(SD)$ using (10.47). This entire process was also repeated 50 times using 50 independent sets of 1000 data points, at the end of which 50 $\mathrm{RMSE}_{ns2-1}(SD)$ values were obtained. The average values and standard deviations of $\mathrm{RMSE}_{ns2}(SD)$ are also plotted in Fig. 10.1 for each of the six epochs. Observe that:

1. After six epochs of training, there is only a small improvement in performance (in both the mean and standard deviation of the RMSE) for the T1 non-singleton IT2 Mamdani fuzzy system over its singleton IT2 counterpart.

2. The minimum values of both the mean and standard deviation of $\mathrm{RMSE}_{ns2-1}(SD)$ occur at the first epoch, and there is a substantially lower value for the standard deviation of $\mathrm{RMSE}_{ns2-1}(SD)$ at the first epoch than the value of the standard deviation of $\mathrm{RMSE}_{s2}(SD)$.

The last observation *strongly* suggests that the T1 non-singleton IT2 Mamdani fuzzy system can be used in a real-time adaptive environment.

As mentioned in the first paragraph of Sect. 10.3.1, the T1 non-singleton IT2 fuzzy system forecaster represents, for the first time, a fuzzy system that accounts for *all* of the uncertainties that are present, namely, rule uncertainties due to training with noisy data and measurement uncertainties due to noisy measurements that are used during actual forecasting. Accounting for all the sources of uncertainties has led to a design that performs almost optimally after just one epoch of training. The earlier designs, which did not account for all of the sources of uncertainties, do not have this property.[6]

The next and final level of uncertainty occurs when the additive measurement noise is nonstationary. In this case, modeling the measurements as IT2 fuzzy numbers is more appropriate than modeling them as T1 fuzzy numbers.

10.3.2 Forecasting of Time Series When the Measurement Noise Is Nonstationary

In actual time-series, such as the price curve for the US. dollar versus the German mark, market volatility can change noticeably over the course of time, so the variance of the noise component, which is related to volatility, need not be constant (Magdon-Ismail et al. 1998). In that case, the additive measurement noise is nonstationary. This section assumes that measurements of a time-series are corrupted by additive zero-mean noise, whose SNR varies in an unknown manner from 0 dB (with standard deviation $\sigma_{n_{0\,\mathrm{dB}}}$) to 10 dB (with standard deviation $\sigma_{n_{10\,\mathrm{dB}}}$).

To begin, an explanation is due as to why an IT2 non-singleton IT2 fuzzy system is appropriate for the case of nonstationary additive noise. Recall that Sect. 4.3.5 provided the following (in-line) well-known formula for SNR:

[6]Cara et al. (2013) compare a singleton IT2 fuzzy system with a non-singleton T1 fuzzy system for nine function approximation problems to address a criticism of IT2 fuzzy systems that they have more parameters (design degrees of freedom) than a T1 fuzzy system. Both fuzzy systems are constrained to have the same number of parameters. The IT2 fuzzy system uses the FOU to handle input uncertainties, whereas the T1 fuzzy system uses non-singleton fuzzification to do this. The parameters of both kinds of fuzzy system are optimized. The paper demonstrates that the IT2 fuzzy system gives much better performance than the non-singleton T1 fuzzy system for high noise levels, and that it is not the use of extra parameters that lets a singleton IT2 fuzzy system outperform a non-singleton T1 fuzzy system, but rather their different ways of handing such uncertainty.

$$\mathrm{SNR} = 10\log_{10}\left(\sigma_s^2/\sigma_n^2\right) \tag{10.48}$$

where σ_s^2 is the variance of the signal and σ_n^2 is the variance of the noise, after which σ_n was solved for, as

$$\sigma_n = \frac{\sigma_s}{10^{\mathrm{SNR}/20}} \tag{10.49}$$

There are two sources of uncertainty in this equation, σ_s and SNR. Because the signal is not known ahead of time—it is being estimated—σ_s is unknown, and all that is known about SNR is that it varies over a range of values. These uncertainties suggest that a useful way to model each input measurement in the IT2 fuzzy system is as a Gaussian that is centered at the measurement and whose standard deviation varies over an interval of values; i.e., as an IT2 fuzzy number.

In the rest of this section, a number of different designs of IT2 non-singleton IT2 Mamdani fuzzy system forecasters are examined.

10.3.2.1 Six-Epoch Derivative-Based Design

Here the designs of five fuzzy system forecasters are compared for the Mackey–Glass time-series. All are Mamdani fuzzy systems, and are: singleton T1, non-singleton T1, singleton IT2, T1 non-singleton IT2, and IT2 non-singleton IT2. All designs use steepest descent and are based on 1000 *noisy* data points: $x(1001), x(1002), \ldots, x(2000)$, where $x(1001), x(1002), \ldots, x(1504)$ are used for training (i.e., for the designs of the fuzzy system forecasters) whereas $x(1505)$, $x(1506), \ldots, x(2000)$ are used for testing the designs. The noise-free Mackey-Glass time-series is the same one that was used in Sect. 10.3.1 (see Fig. 4.4), but now its sampled values, $s(k)$, are corrupted by uniformly distributed nonstationary additive noise, $n(k)$, so that $x(k)$ is given as in (10.45), where $0\,\mathrm{dB} \le \mathrm{SNR} \le 10\,\mathrm{dB}$ so that $\sigma_{n_{0\,\mathrm{dB}}} \le \sigma_n \le \sigma_{n_{10\,\mathrm{dB}}}$. At each value of k, σ_n is uniformly distributed in the interval $[\sigma_{n_{0\,\mathrm{dB}}}, \sigma_{n_{10\,\mathrm{dB}}}]$, which was broken into 100 levels. One realization of $x(1001), x(1002), \ldots, x(2000)$ is depicted in Fig. 10.2. It may be interesting for the reader to compare Figs. 10.2 and 4.4. The latter is also one realization of $x(1001)$, $x(1002), \ldots, x(2000)$ but for stationary noise at 0 dB. The differences between the two noisy time-series are quite noticeable.

Once again, rules with the four antecedents $x(k-3)$, $x(k-2)$, $x(k-1)$ and $x(k)$ were used to predict $x(k+1)$, and two fuzzy sets were used for each antecedent; hence, there were 16 rules. Gaussian MFs were chosen for the antecedents of the two T1 fuzzy systems, and Gaussian primary MFs of uncertain means were chosen for the antecedents of the three IT2 fuzzy systems. Gaussian MFs were chosen for the input measurements in the T1 non-singleton cases, whereas a Gaussian primary MF with uncertain standard deviation was chosen for the input measurements in the IT2 non-singleton case. The parameters as well as the number

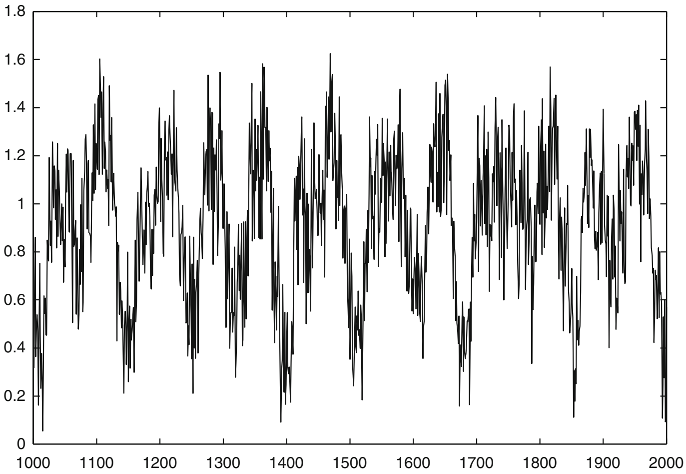


Fig. 10.2 One realization of uniformly distributed nonstationary noisy data, $x(1001)$, $x(1002)$, $\ldots$, $x(2000)$, where $\sigma_{n_{0\,\text{dB}}} \leq \sigma_n \leq \sigma_{n_{10\,\text{dB}}}$

of parameters that characterize each of the 5 fuzzy systems are summarized in Table 10.6. To use the results in that table, set $p = 4$ and $M = 16$.

All five designs used the *totally independent approach* that was described toward the end of Sect. 10.1; i.e., all of the parameters were tuned independently for each design.

Initial values for the parameters in each of the five designs are summarized in Table 10.7. Note that m_x and σ_x are the mean and standard deviation, respectively, of the data in the 504 training samples, $x(1001)$, $x(1002)$, …, $x(1504)$, and

$$\sigma_n \equiv (\hat{\sigma}_{n_{0\,\text{dB}}} + \hat{\sigma}_{n_{10\,\text{dB}}})/2 \tag{10.50}$$

where $\hat{\sigma}_{n_{0\,\text{dB}}}$ and $\hat{\sigma}_{n_{10\,\text{dB}}}$ are guesstimates of $\sigma_{n_{0\,\text{dB}}}$ and $\sigma_{n_{10\,\text{dB}}}$, respectively.

Table 10.6 The parameters and numbers of parameters in five different fuzzy systems

Fuzzy system	P in one input set	P in one antecedent	P in one consequent	Total # of P
S T1	NA	$m_{F_k^i}, \sigma_{F_k^i}$	$\bar{y}^i$	$2pM + M$
NS T1	σ_{X_k}	$m_{F_k^i}, \sigma_{F_k^i}$	$\bar{y}^i$	$2pM + M + p$
S IT2	NA	$m_{k1}^i, m_{k2}^i, \sigma_k^i$	y_l^i, y_r^i	$3pM + 2\,M$
T1 NS IT2	σ_{X_k}	$m_{k1}^i, m_{k2}^i, \sigma_k^i$	y_l^i, y_r^i	$3pM + 2\,M + p$
IT2 NS IT2	σ_{k1}, σ_{k2}	$m_{k1}^i, m_{k2}^i, \sigma_k^i$	y_l^i, y_r^i	$3pM + 2\,M + 2p$

(with M rules and p antecedents in each rule, i.e., $i = 1, \ldots, M$ and $k = 1, \ldots, p$); P is short for *parameters*

Table 10.7 *Initial values* of the parameters in five fuzzy systems, where each antecedent is described by two fuzzy sets $(i = 1, \ldots, 16 \text{ and } k = 1, \ldots, 4)$

Fuzzy system	Input	For each antecedent	Consequent
S T1	NA	Mean: $m_x - 2\sigma_x$ or $m_x + 2\sigma_x$, $\sigma_{F_k^i} = 2\sigma_x$	$\bar{y}^i \in [0, 1]$
NS T1	$\sigma_{X_k} = \sigma_n$	Mean: $m_x - 2\sigma_x$ or $m_x + 2\sigma_x$, $\sigma_{F_k^i} = 2\sigma_x$	$\bar{y}^i \in [0, 1]$
S IT2	NA	Mean: $[m_x - 2\sigma_x - 0.25\sigma_n, m_x - 2\sigma_x + 0.25\sigma_n]$ or $[m_x + 2\sigma_x - 0.25\sigma_n, m_x + 2\sigma_x + 0.25\sigma_n]$ $\sigma_k^i = 2\sigma_x$	$y_l^i = \bar{y}^i - \sigma_n$ $y_r^i = \bar{y}^i + \sigma_n$
T1 NS IT2	$\sigma_{X_k} = \sigma_n$	Mean: $[m_x - 2\sigma_x - 0.25\sigma_n, m_x - 2\sigma_x + 0.25\sigma_n]$ or $[m_x + 2\sigma_x - 0.25\sigma_n, m_x + 2\sigma_x + 0.25\sigma_n]$ $\sigma_k^i = 2\sigma_x$	$y_l^i = \bar{y}^i - \sigma_n$ $y_r^i = \bar{y}^i + \sigma_n$
IT2 NS IT2	$\sigma_{k1} = \hat{\sigma}_{n_{10\,\mathrm{dB}}}$ $\sigma_{k2} = \hat{\sigma}_{n_{0\,\mathrm{dB}}}$	Mean: $[m_x - 2\sigma_x - 0.25\sigma_n, m_x - 2\sigma_x + 0.25\sigma_n]$ or $[m_x + 2\sigma_x - 0.25\sigma_n, m_x + 2\sigma_x + 0.25\sigma_n]$ $\sigma_k^i = 2\sigma_x$	$y_l^i = \bar{y}^i - \sigma_n$ $y_r^i = \bar{y}^i + \sigma_n$

The performance of all the designs were again evaluated using the respective RMSEs in (4.51), (4.53), (10.46) and (10.47), as well as the following RMSE for the IT2 non-singleton IT2 design:

$$\mathrm{RMSE}_{ns2-2}(SD) = \sqrt{\frac{1}{496}\sum_{k=1504}^{1999}\left[s(k+1) - y_{ns2-2}(\mathbf{x}^{(k)})\right]^2} \tag{10.51}$$

In (10.51), y_{ns2-2} was computed using (9.113), (9.112) and (9.111), where the firing interval was computed using (9.82) and (9.83).

Each fuzzy system was tuned using a steepest descent algorithm in which the learning parameter $\beta_\theta = 0.4$. Training and testing were carried out for six epochs. After each epoch the testing data was used to see how each fuzzy system performed, by computing $\mathrm{RMSE}_{s1}(SD)$, $\mathrm{RMSE}_{ns1}(SD)$, $\mathrm{RMSE}_{s2}(SD)$, $\mathrm{RMSE}_{ns2-1}(SD)$ and $\mathrm{RMSE}_{ns2-2}(SD)$. This entire process was repeated 50 times using 50 independent sets of 1000 data points, at the end of which 50 of the 5 $\mathrm{RMSE}(SD)$ values were obtained. The average values and standard deviations of these RMSEs are plotted in Fig. 10.3 for each of the six epochs. Observe that:

1. IT2 fuzzy systems outperform the T1 fuzzy systems. The IT2 non-singleton IT2 fuzzy system performs the best and the T1 non-singleton IT2 fuzzy system also gives very good results. The reason for the latter is because the T1 non-singleton IT2 fuzzy system used σ_n in (10.50) as the initial value for the standard deviation of its input measurement MFs, and this value of σ_n gives a good approximation to the average value of the standard deviation of the uniform noise.
2. The non-singleton IT2 fuzzy systems achieve close to their optimal performance almost at the first epoch of tuning. This shows that non-singleton IT2 fuzzy systems (as compared to T1 fuzzy systems) are very promising for real-time signal processing, where more than one epoch of tuning is not possible.

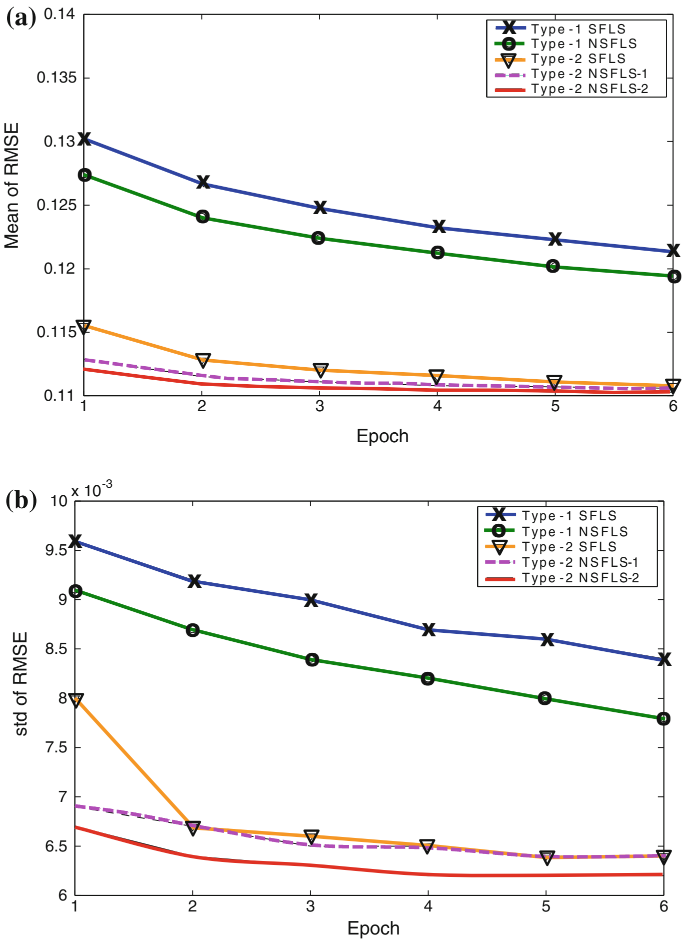


Fig. 10.3 The mean and standard deviation of $\text{RMSE}_{s1}(SD)$, $\text{RMSE}_{ns1}(SD)$, $\text{RMSE}_{s2}(SD)$, $\text{RMSE}_{ns\,2-1}(SD)$ and $\text{RMSE}_{ns\,2-2}(SD)$, averaged over 50 Monte Carlo designs. Tuning was performed in each realization for six epochs: **a** mean values, and **b** standard deviation values. Note that "FLS" is equivalent to "fuzzy system"

3. The standard deviations of the RMSEs show that the IT2 fuzzy systems (especially the T1 and IT2 non-singleton IT2 fuzzy systems) have a considerably smaller standard deviation than do the T1 fuzzy systems, which demonstrates that IT2 fuzzy systems are much more robust to the nonstationary noise than are T1 fuzzy systems. Hence, IT2 fuzzy systems appear to be promising for use in adaptive filters, such as channel equalizers (e.g., Liang and Mendel 2000a; Patra and Mulgrew 1998; Sarwal and Srinath 1995 and Wang and Mendel 1993), because such equalizers must be robust to additive noise (see Sect. 10.7).

10.3.2.2 One-Epoch Combined Derivative-Based and SVD-QR Design

In this fuzzy system design, the derivative-based (steepest descent) and SVD–QR methods were combined. To do this the steepest descent method was used for just one epoch of training after which the SVD–QR method was applied to its results. As in the previous section, five Mamdani fuzzy system forecasters were designed: singleton T1, non-singleton T1, singleton IT2, T1 non-singleton IT2, and IT2 non-singleton IT2. All of the previous section's discussions about number of data points, training points, testing points, number of rule antecedents, number of fuzzy sets for each antecedent, number of rules, choices for antecedent, consequent and input measurement MFs, initial choices for MF parameters (using the totally independent design approach), and evaluation by means of RMSE formulas remain the same for the present designs.

50 Monte Carlo realizations were run for each of the five designs, and for each realization the fuzzy system was tuned before rule-reduction using a simple steepest descent algorithm, but only for one epoch. Each fuzzy system was then rule-reduced using the appropriate SVD–QR method (see discussions about SVD–QR designs in Sects. 4.2.4 and 10.2.4). The number of rules to be retained was established by using a threshold, γ (set arbitrarily to 1), for the singular values that were computed for the SVD of a FBF matrix [e.g., (10.29) and (10.30), making use of the discussions on how to use these FBF matrices for an IT2 Mamdani fuzzy system with COS type-reduction + defuzzification, that is, given at the end of Example 10.9]. Let s_j denote those singular values; then $\hat{r}$ was chosen such that $s_{\hat{r}} \geq 1$.

RMSEs were computed both before and after rule-reduction. Results are summarized in Tables 10.8 and 10.9. Observe, from Table 10.9 that there is a very substantial reduction in the number of rules, from 16 to anywhere from 4 to 9. Unfortunately, there is an accompanying degradation in RMSE performance, as can be seen from the entries in Table 10.8. Our next design attempts to both improve the rule-reduced RMSEs and to further reduce the number of rules.

10.3.2.3 Six-Epoch Iterative Combined Derivative-Based and SVD-QR Design

Next, the designs of five fuzzy system forecasters are compared for the Mackey–Glass time-series using an iterative version of combined derivative-based and SVD–QR methods. As in the previous sections, five Mamdani fuzzy system forecasters were designed: singleton T1, non-singleton T1, singleton IT2, T1 non-singleton IT2, and IT2 non-singleton IT2. Additionally, all of Sect. 10.3.2.1's discussions about number of data points, training points, testing points, number of rule antecedents, number of fuzzy sets for each antecedent, number of rules, choices for antecedent, consequent and input measurement MFs, initial choices for MF parameters (using the totally independent design approach), and evaluation by means of RMSE formulas remain the same for the present designs.

Table 10.8 Mean and standard deviation (SD) values for RMSEs (for the test data) for the five fuzzy system designs averaged over 50 Monte Carlo realizations

Mean and SD of RMSE for T1 fuzzy systems				
Rules	Singleton		Non-singleton	
	Mean	SD	Mean	SD
16 rules	0.1323	0.0098	0.1287	0.0088
$\hat{r}$ rules	0.1626	0.0272	0.1674	0.0295

Mean and SD of RMSE for IT2 fuzzy system						
Rules	Singleton		TI non-singleton		IT2 non-singleton	
	Mean	SD	Mean	SD	Mean	SD
16 rules	0.1154	0.0082	0.1132	0.0071	0.1130	0.0070
$\hat{r}$ rules	0.1536	0.0261	0.1519	0.0256	0.1509	0.0246

Tuning was performed in each realization for just one epoch. Information about $\hat{r}$ is given in Table 10.9

Table 10.9 Number of reduced rules: their range, mean, and standard deviations (SD)

Number of reduced rules for T1 fuzzy systems					
Singleton			Non-singleton		
Range	Mean	SD	Range	Mean	SD
[4, 6]	5.2	0.6389	[4, 6]	5.12	0.6273

Number of reduced rules for IT2 fuzzy systems								
Singleton			TI non-singleton			IT2 non-singleton		
Range	Mean	SD	Range	Mean	SD	Range	Mean	SD
[4, 9]	6.18	1.1726	[4, 9]	6.2	1.1066	[4, 9]	6.22	1.2171

During epoch #1 the parameters of a 16-rule fuzzy system were tuned, after which the 16 rules were reduced to $\hat{r}_1$ rules using the appropriate SVD–QR method. As in the previous design, a threshold of 1 was used for the singular values and $\hat{r}$ was chosen such that $s_{\hat{r}} \geq 1$. During epoch #2 the parameters of the $\hat{r}_1$ rule fuzzy system were tuned after which the $\hat{r}_1$ rules were reduced to $\hat{r}_2$ rules, again using the appropriate SVD–QR method with a threshold of 1 for the singular values. This was continued for six epochs, and after six epochs, a fuzzy system was obtained that had $\hat{r}_6$ rules.

The mean and standard deviation of the five-rule-reduced RMSEs for each epoch and each design are summarized in Fig. 10.4a, b, and the mean and standard deviation of the number of rules for each epoch and design in Fig. 10.5a, b. Observe, from these figures that:

1. Combining steepest descent and SVD–QR iteratively has indeed improved the rule-reduced RMSE.
2. The rule-reduced IT2 fuzzy systems all perform better than the rule-reduced T1 fuzzy systems. Additionally, the former are more robust than the latter because of their lower RMSE standard deviations.

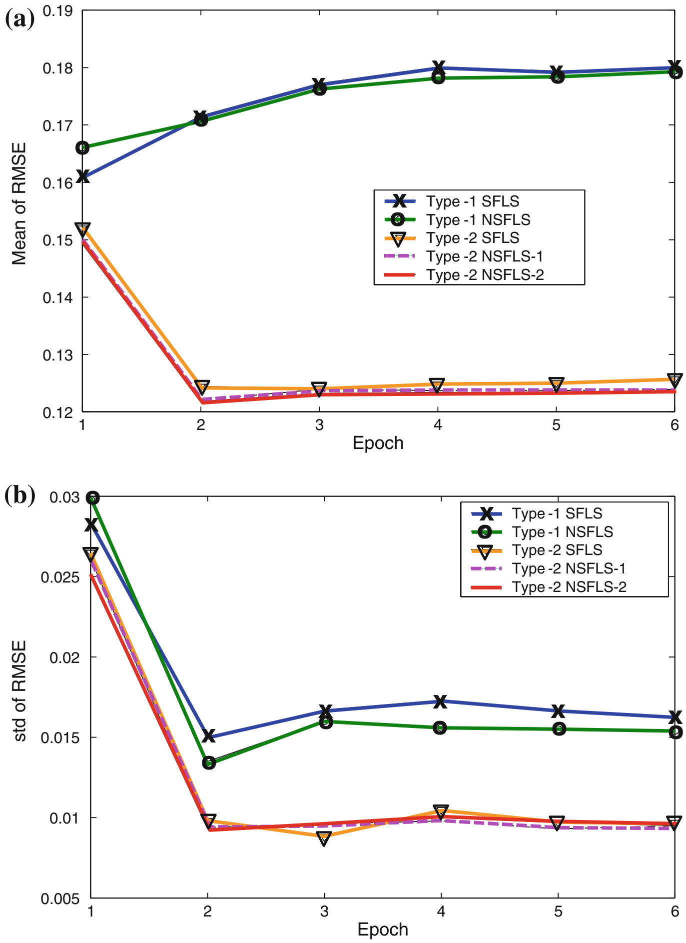


Fig. 10.4 The mean and standard deviation of the rule-reduced RMSEs (for the test data) for the five fuzzy system designs averaged over 50 Monte Carlo realizations. Tuning was performed in each realization for six epochs; **a** mean values, and **b** standard deviations. Note that "FLS" is equivalent to "fuzzy system"

3. The rule-reduced IT2 fuzzy system RMSEs either decrease or increase slightly from epoch-to-epoch, as the average number of their rules decreases from epoch-to-epoch. Meanwhile, rule-reduced T1 fuzzy systems RMSEs increase from epoch-to-epoch, as the average number of their rules decreases from epoch-to-epoch.
4. A rule-reduced IT2 fuzzy system has, on average, one more rule than does a rule-reduced T1 fuzzy system after the six epochs. However, comparing the performance difference of the rule-reduced IT2 fuzzy system to that of the rule-reduced T1 fuzzy system, this gain seems to be worthwhile.

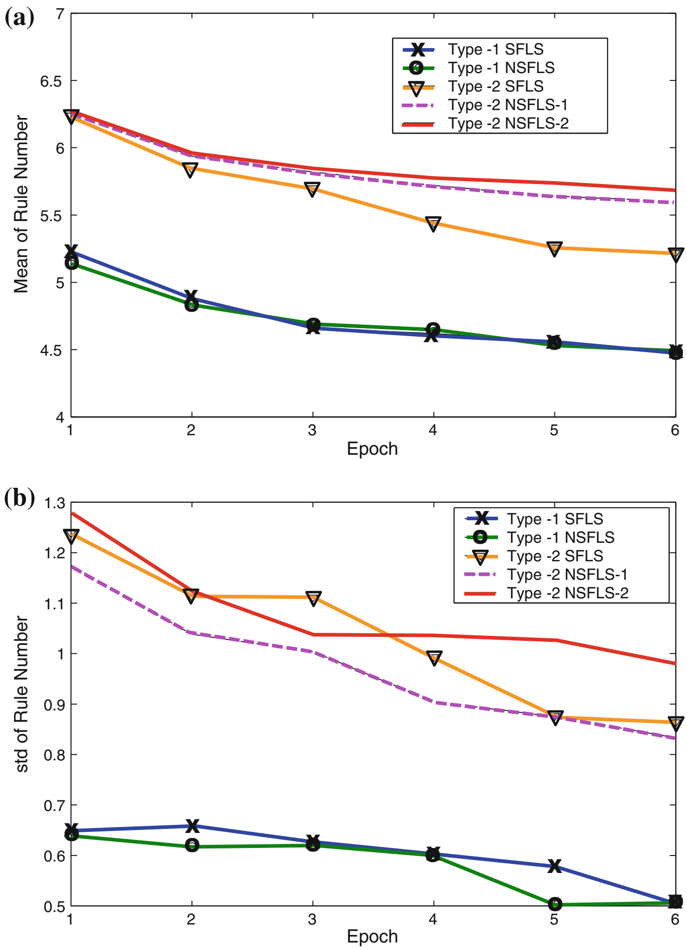


Fig. 10.5 The mean and standard deviation of the number of rules in each epoch, after rule-reduction, using combined derivative-based tuning and an SVD–QR method. Note that "FLS" is equivalent to "fuzzy system"

5. The final RMSE performances of the three IT2 fuzzy systems are about the same. Hence, our choice would be to implement the simpler of these fuzzy systems, namely the singleton IT2 fuzzy system.

This last design has demonstrated that the initial number of rules—16—can be greatly reduced, but with a small loss of overall RMSE performance (compare the six-epoch RMSE values in Fig. 10.4 with those in Fig. 10.3). Obviously, there is a tradeoff between accuracy and complexity to achieve rule-reduction through design.[7]

For this final forecasting case study, in which the additive noise was nonstationary, it is clear that the IT2 fuzzy system models better represent this situation than do the T1 fuzzy system models.

10.4 Case Study: Knowledge Mining Using Surveys

The problem of knowledge mining using surveys has been described in Sect. 4.4. Here explanations and demonstrations are provided for how to design a singleton IT2 Fuzzy Logic Advisor (FLA) using the data presented in Tables 4.10–4.13. Wherever possible, the presentations in Sect. 4.4 are paralleled so that the reader can compare the T1 and IT2 FLAs. It is advisable for the reader to review Sect. 4.4.1 because it provides the methodology for knowledge mining using surveys that is followed in this section. Without going into details, the methodology is:

1. Identify the behavior of interest.
2. Determine the indicators of the behavior of interest.
3. Establish scales for each indicator and the behavior of interest.
4. Establish names and interval information for each of the indicator's fuzzy sets and behavior of interest's fuzzy sets.
5. Establish the rules.
6. Survey people (experts) to provide consequents for the rules.

10.4.1 Determining the IT2 FSs for the Vocabulary

First, IT2 MFs must be associated with the Table 4.10 interval data. To do this one must decide on an FOU for each of the MFs. This would be a good time for the reader to review the procedure that was used in Sect. 4.4.3 to construct the MFs depicted in Fig. 4.8. Recall that Sect. 4.4.3 stated:

> There is uncertainty associated with our use of $m_a - \sigma_a$ and $m_b + \sigma_b$; e.g., why not use $m_a - 0.5\sigma_a$ or $m_a - 2\sigma_a$ instead of $m_a - \sigma_a$, and $m_b + 0.5\sigma_b$ or $m_b + 1.5\sigma_b$ instead of $m_b + \sigma_b$? This uncertainty cannot be captured using T1 FSs; however, as is explained in Sect. 10.4.1, it can be captured using T2 FSs.

This uncertainty is captured in an IT2 FS by using a *FOU*, as is explained next.

[7]If the singular value threshold γ had been chosen to be smaller than 1, then the rule-reduced designs would have contained more rules, but would have achieved even smaller RMSEs than in the present designs.

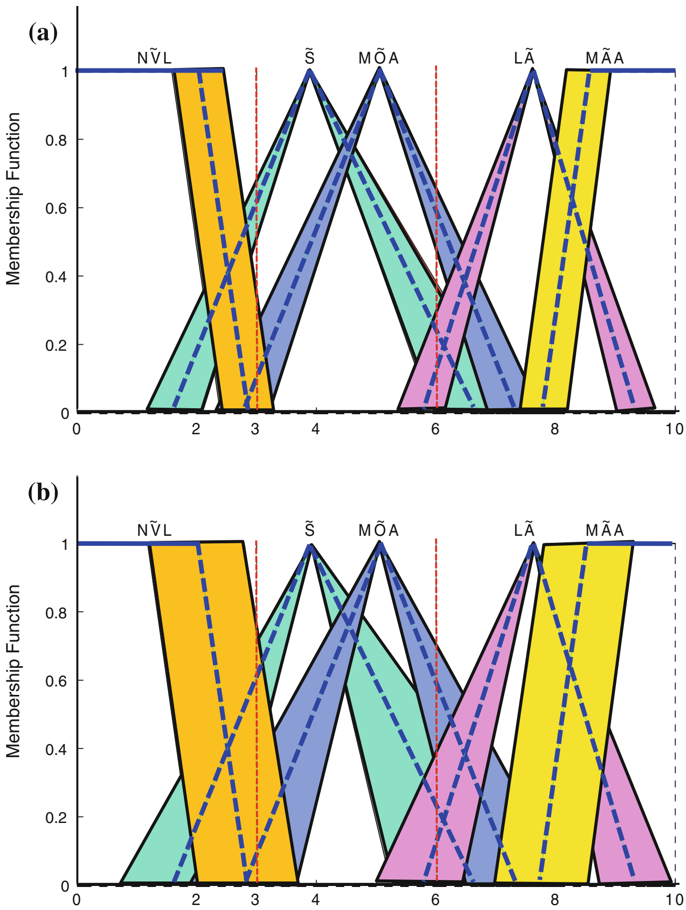


Fig. 10.6 The FOUs used to represent the five linguistic labels. The *dashed lines* denote the T1 MFs (see Fig. 4.8) that were used to represent the five linguistic labels. **a** $\rho = 0.5$ (50% uncertainty), and **b** $\rho = 1$ (100% uncertainty)

The same linguistic labels are used for the IT2 FSs as were used in Sect. 4.4.3 for their type-1 counterparts, namely: $\tilde{F}_1$ = *none to very little* (*NVL*), $\tilde{F}_2$ = *some* (*S*), $\tilde{F}_3$ = *a moderate amount* (*MOA*), $\tilde{F}_4$ = *a large amount* (*LA*), and $\tilde{F}_5$ = *a maximum amount* (*MAA*). Triangle principal MFs (shown dashed in Fig. 10.6) are used for the three interior labels *some*, *a moderate amount*, and *a large amount*, and piecewise-linear principal MFs (also shown dashed in Fig. 10.6) are used for the leftmost and rightmost labels *none to very little* and *a maximum amount*. These choices were made in Mendel (2001) to be consistent with the choices made in Sect. 4.4.3, and because it is relatively easy, as is explained next, to go from the data in Table 4.10 to FOUs for each of the principal MFs. See Sect. 10.4.6 for discussions on some other ways to map interval endpoint data that are collected from a group of subjects into an FOU.

The FOUs were obtained by specifying upper and lower MFs for each fuzzy set. Let ρ denote a *fraction of uncertainty*; i.e., $0 \leq \rho \leq 1$. Then, each FOU was constructed as follows (see Table 4.10 for values of m_a, m_b, σ_a, and σ_b):

1. For the *triangle interior FOUs*, the coordinates of the UMF are at $(m_a - (1+\rho)\sigma_a, 0)$, $((m_a + m_b)/2, 1)$ and $(m_b + (1+\rho)\sigma_b, 0)$, and the coordinates of the LMF are at $(m_a - (1-\rho)\sigma_a, 0)$, $((m_a + m_b)/2, 1)$, and $(m_b + (1-\rho)\sigma_b, 0)$.
2. For the *piecewise-linear shoulder FOUs*, the coordinates of the UMF are at[8] $(m_a - (1+\rho)\sigma_a, 0)$, $(m_a - \rho\sigma_a, 1)$, $(m_b + \rho\sigma_b, 1)$, and $(m_b + (1+\rho)\sigma_b, 0)$, and the coordinates of the LMF are at $(m_a - (1-\rho)\sigma_a, 0)$, $(m_a + \rho\sigma_a, 1)$, $(m_b - \rho\sigma_b, 1)$, and $(m_b + (1-\rho)\sigma_b, 0)$. Note that the slopes of the legs of the piecewise-linear MFs have been chosen arbitrarily.

Figure 10.6a depicts the FOUs for $\rho = 0.5$ (which corresponds to a 50% level of uncertainty) and Fig. 10.6b depicts the FOUs for $\rho = 1$ (which corresponds to a 100% level of uncertainty). The Fig. 10.6 FOUs are used for both the antecedents and consequents of FLA rules. When $\rho = 0$, the Fig. 10.6 FOUs reduce to the MFs in Fig. 4.8.

10.4.2 What Does One Do with a Histogram of Responses?

Results from three rule surveys are summarized in Tables 4.11–4.13. This section associates rules with these surveys in which their antecedents and consequents are IT2 FSs; i.e., the two single-antecedent surveys are associated with questions of the form "IF x_1 is $\tilde{F}^i$, THEN what is y?" $(i = 1, \ldots, 5)$, or "IF x_2 is $\tilde{F}^j$, THEN what is y?"$(j = 1, \ldots, 5)$, whereas the two-antecedent survey is associated with questions of the form " IF x_1 is $\tilde{F}^i$ and x_2 is $\tilde{F}^j$, THEN what is y"$(i, j = 1, \ldots, 5)$. Note that, regardless of how one models the antecedents and consequents in the fuzzy logic advisors, to the people who answered the survey questions, "a rule is a rule is a rule."

Tables 4.11–4.13 reveal that each rule has a histogram of responses. Section 4.4.4 raised the question "What should be done with this information?" and described three possibilities: Keep the response chosen by the largest number of experts, find a weighted average of the rule consequents for each rule, or preserve the distributions of the expert-responses for each rule. For reasons explained in Sect. 4.4.4 the second possibility was chosen. The same is done here; i.e., *the responses are averaged.*

---

[8] Note that, for $\tilde{NVL}$, $m_a = \sigma_a = 0$, so the first two coordinates of both its UMF and LMF are (0, 0) and (0, 1), as they should be; and, for $\tilde{MAA}$, $m_b = 10$ and $\sigma_b = 0$, so the last two coordinates of both its UMF and LMF are (10, 1) and (10, 0), as they should be.

Table 10.10 Centroids, $C_{\tilde{G}_i}$, of consequent IT2 FSs, computed using EIASC or EKM algorithms

Consequent IT2 FS	Centroid[a]	Figure 10.6a centroid ($\rho = 0.5$)	Figure 10.6b centroid ($\rho = 1$)
$NVL\,(\tilde{G}_1)$	$C_{\tilde{G}_1}$	[0.9808, 1.3820]	[0.7817, 1.5881]
$S\,(\tilde{G}_2)$	$C_{\tilde{G}_2}$	[3.6717, 4.4320]	[3.2856, 4.8338]
$MOA\,(\tilde{G}_3)$	$C_{\tilde{G}_3}$	[4.7501, 5.3314]	[4.4545, 5.6260]
$LA\,(\tilde{G}_4)$	$C_{\tilde{G}_4}$	[7.3418, 7.7915]	[7.1125, 8.0184]
$MAA\,(\tilde{G}_5)$	$C_{\tilde{G}_5}$	[8.9159, 9.2842]	[8.7136, 9.4651]

[a]Only the support of the centroid is given in this table

When this is done, the consequent of each rule is treated as a T1 FS, C^l_{avg} (e.g., a two-antecedent rule is now interpreted as "IF x_1 is $\tilde{F}^i$ and x_2 is $\tilde{F}^j$, THEN $y = C^l_{\text{avg}}$"), where

$$C^l_{\text{avg}} = \frac{\sum_{i=1}^{M} w^l_i C_{\tilde{G}_i}}{\sum_{i=1}^{M} w^l_i} \equiv 1/\left[\underline{C}^l_{\text{avg}}, \bar{C}^l_{\text{avg}}\right] \tag{10.52}$$

in which the top sum is an algebraic sum (of type-1 interval fuzzy numbers), $C_{\tilde{G}_i}$ is the centroid of the ith consequent set (given in Table 10.10), and w^l_i is the weight associated with the ith consequent for the lth rule. The entries in Tables 4.11–4.13 are the weights. The computation in (10.52) is easy to perform and is done using Example 2.22. Tables 10.11–10.13 summarize these calculations and are analogous to Tables 4.11–4.13. Comparing the C^l_{avg} intervals in these tables to their respective c^l_{avg} values in Tables 4.11–4.13, observe that the latter always fall within the former, as expected.

10.4.3 IT2 Consensus FLAs

Equations (4.57)–(4.59) describe the three consensus T1 FLAs, $y_{c1}(x_1)$, $y_{c1}(x_2)$, and $y_{c1}(x_1, x_2)$. Their IT2 counterparts (IT2 Mamdani fuzzy systems with COS type-reduction + defuzzification) are obtained using (9.172)–(9.176), and for completeness, are stated next.[9]

[9]Note that $\underline{C}^i_{avg}$ and $\bar{C}^i_{avg}$ play the roles of $c_l(\tilde{G}^i)$ and $c_r(\tilde{G}^i)$, respectively.

Table 10.11 Histogram of survey responses for single-antecedent rules between indicator x_1 and consequent y

Rule no.	Consequent					C^l_{avg} [a]
	NVL	S	MOA	LA	MAA	
1 (NVL)	42	3	2	0	0	[1.3130, 1.7447]
2 (S)	33	12	0	2	0	[1.9385, 2.4335]
3 (MOA)	12	16	15	3	1	[3.6747, 4.2580]
4 (LA)	3	6	11	25	2	[5.9277, 6.4413]
5 (MAA)	3	6	8	22	8	[6.2941, 6.7888]

Entries denote the number of respondents out of 47 (w^l_i) that chose the consequent. C^l_{avg} is the weighted average of the responses, given by (10.52). For notational simplicity, the tildes have been omitted over the fuzzy set labels

[a] Only the support of the weighted average is given in this table

Table 10.12 Histogram of survey responses for single-antecedent rules between indicator x_2 and consequent y

Rule no.	Consequent					C^l_{avg} [a]
	NVL	S	MOA	LA	MAA	
1 (NVL)	36	7	4	0	0	[1.7024, 2.1724]
2 (S)	26	17	4	0	0	[2.2749, 2.8213]
3 (MOA)	2	16	27	2	0	[4.3328, 4.9618]
4 (LA)	1	3	11	22	10	[6.7006, 7.1825]
5 (MAA)	0	3	7	17	20	[7.3914, 7.8458]

Entries denote the number of respondents out of 47 (w^l_i) that chose the consequent. C^l_{avg} is the weighted average of the responses, given by (10.52). For notational simplicity, the tildes have been omitted over the fuzzy set labels

[a]Only the support of the weighted average is given in this table

- *Consensus IT2 FLA for* x_1:

$$y^{\text{COS}}_{c2}(x_1) = \frac{1}{2} y^{\text{COS}}_{c2,l}(x_1) + \frac{1}{2} y^{\text{COS}}_{c2,r}(x_1) = \frac{1}{2} \sum_{i=1}^{5} \underline{C}^i_{\text{avg}} \phi^i_{c2,l}(x_1) + \frac{1}{2} \sum_{i=1}^{5} \bar{C}^i_{\text{avg}} \phi^i_{c2,r}(x_1) \tag{10.53}$$

where $(i = 1, \ldots, 5)$ $\underline{C}^i_{\text{avg}}$ and $\bar{C}^i_{\text{avg}}$ are given in the last column of Table 10.11, and:

$$\phi^i_{c2,l}(x_1) = \frac{\delta^i_{c2,l} \bar{f}^i(x_1) + (1 - \delta^i_{c2,l}) \underline{f}^i(x_1)}{\sum_{i=1}^{5} [\delta^i_{c2,l} \bar{f}^i(x_1) + (1 - \delta^i_{c2,l}) \underline{f}^i(x_1)]} \tag{10.54}$$

Table 10.13 Histogram of survey responses for two-antecedent rules between indicators x_1 and x_2 and consequent y

Rule no.	Consequent					C^l_{avg} [a]
	NVL	S	MOA	LA	MAA	
1 (NVL/NVL)	38	7	2	0	0	[1.5420, 2.0043]
2 (NVL/S)	33	11	3	0	0	[1.8512, 2.3479]
3 (NVL/MOA)	6	21	16	4	0	[4.0076, 4.6347]
4 (NVL/LA)	0	12	26	8	1	[5.0045, 5.6046]
5 (NVL/MAA)	0	9	16	19	3	[5.8572, 6.4060]
6 (S/NVL)	31	11	4	1	0	[2.0667, 2.5683]
7 (S/S)	17	23	7	0	0	[2.8590, 3.4628]
8 (S/MOA)	0	19	19	8	1	[4.8439, 5.4706]
9 (S/LA)	1	8	23	13	2	[5.3805, 5.9429]
10 (S/MAA)	0	7	17	21	2	[5.9247, 6.4648]
11 (MOA/NVL)	7	23	16	1	0	[3.7161, 4.3554]
12 (MOA/S)	5	22	20	0	0	[3.8443, 4.4902]
13 (MOA/MOA)	2	7	22	15	1	[5.3449, 5.8986]
14 (MOA/LA)	1	4	13	17	12	[6.5792, 7.0699]
15 (MOA/MAA)	0	4	12	24	7	[6.6022, 7.0998]
16 (LA/NVL)	7	13	21	6	0	[4.2213, 4.8085]
17 (LA/S)	3	11	23	10	0	[4.4085, 5.3922]
18 (LA/MOA)	0	3	18	18	8	[6.3829, 6.8890]
19 (LA/LA)	0	1	9	17	20	[7.4373, 7.8841]
20 (LA/MAA)	1	2	6	11	27	[7.6237, 8.0556]
21 (MAA/NVL)	2	16	18	11	0	[4.8291, 5.4329]
22 (MAA/S)	2	9	22	13	1	[5.1887, 5.7557]
23 (MAA/MOA)	0	3	15	18	11	[6.6488, 7.1413]
24 (MAA/LA)	0	1	7	17	22	[7.6145, 8.0523]
25 (MAA/MAA)	0	2	3	12	30	[8.0250, 8.4443]

Entries denote the number of respondents out of 47 (w^l_i) that chose the consequent. C^l_{avg} is the weighted average of the responses, given by (10.52). For notational simplicity, the tildes have been omitted over the fuzzy set labels

[a]Only the support of the weighted average is given in this table

$$\phi^i_{c2,r}(x_1) = \frac{\delta^i_{c2,r}\bar{f}^i(x_1) + (1 - \delta^i_{c2,r})\underline{f}^i(x_1)}{\sum_{i=1}^{5}[\delta^i_{c2,r}\bar{f}^i(x_1) + (1 - \delta^i_{c2,r})\underline{f}^i(x_1)} \tag{10.55}$$

In (10.54)

$$\delta^i_{c2,l} = \begin{cases} 1 & \underline{C}^i_{\text{avg}} \le y^{\text{COS}}_{c2,l}(x_1) \\ 0 & \text{otherwise} \end{cases} \tag{10.56}$$

and in (10.55)

$$\delta^i_{2c,r} = \begin{cases} 1 & \bar{C}^i_{\text{avg}} \ge y^{\text{COS}}_{2c,r}(x_1) \\ 0 & \text{otherwise} \end{cases} \tag{10.57}$$

Formulas for computing the lower and upper firing intervals depend on the fuzzifier, and are not restated here. They can be found in (9.25) and (9.26) for singleton fuzzification. Because $\underline{C}^i_{\text{avg}}$ and $\bar{C}^i_{\text{avg}}$ $(i = 1, \ldots, 5)$ are known, it is possible to compute $y^{\text{COS}}_{c2,l}(x_1)$ and $y^{\text{COS}}_{c2,r}(x_1)$ using EIASC or EKM algorithms, which means that $\delta^i_{c2,l}$ and $\delta^i_{2c,r}$ can be computed, so that (10.53)–(10.57) can actually be used to compute $y_{c2}(x_1)$.

- *Consensus IT2 FLA for* x_2: Use the equations for the *Consensus IT2 FLA for* x_1 in which x_1 is replaced by x_2, and $C^i_{\text{avg}} = [\underline{C}^i_{\text{avg}}, \bar{C}^i_{\text{avg}}]$ is now given in the last column of Table 10.12.
- *Consensus IT2 FLA for* x_1 *and* x_2: Use the equations for the *Consensus IT2 FLA for* x_1 in which: (a) x_1 is replaced by x_1, x_2; (b) there are 25 terms in each of the summations; (c) $\bar{\mu}_{\tilde{F}^i_1}(x_1)$ is replaced by $\bar{\mu}_{\tilde{F}^i_1}(x_1) \times \bar{\mu}_{\tilde{F}^i_2}(x_2)$; (d) $\underline{\mu}_{\tilde{F}^i_1}(x_1)$ is replaced by $\underline{\mu}_{\tilde{F}^i_1}(x_1) \times \underline{\mu}_{\tilde{F}^i_2}(x_2)$; and, (e) $C^i_{\text{avg}} = [\underline{C}^i_{\text{avg}}, \bar{C}^i_{\text{avg}}]$ is now given in the last column of Table 10.13.

Example 10.11 A sample calculation of $y^{\text{COS}}_{c2}(x_1, x_2)$ when $(x_1, x_2) = (3, 6)$ and $\rho = 0.5$ is provided in this example. Observe, from the vertical dashed lines in Fig. 10.6a, that $x_1 = 3$ fires the three IT2 FSs $N\tilde{V}L$, $\tilde{S}$, and $M\tilde{O}A$ and that $x_2 = 6$ fires the three IT2 FSs $\tilde{S}$, $M\tilde{O}A$, and $L\tilde{A}$. It follows, therefore that the following nine rules, whose antecedent pairs are stated, are fired:

$$(N\tilde{V}L, \tilde{S}), (N\tilde{V}L, M\tilde{O}A), (N\tilde{V}L, L\tilde{A}), (\tilde{S}, \tilde{S}), (\tilde{S}, M\tilde{O}A), (\tilde{S}, L\tilde{A}), (M\tilde{O}A, \tilde{S}),$$
$$(M\tilde{O}A, M\tilde{O}A), \text{ and} (M\tilde{O}A, L\tilde{A})$$

From Table 10.13 [note, e.g., that $(NVL/S) = (N\tilde{V}L/\tilde{S})$], observe that these are rules 2, 3, 4, 7, 8, 9, 12, 13, and 14. The firing interval for each of these rules is obtained by using (9.25) and (9.26), and all of them are summarized in Table 10.14. For example, the firing interval, $[\underline{f}^8, \bar{f}^8]$, for $\tilde{R}^8_Z$ is (use the data in Table 10.14 for i = 8)

$$\begin{aligned}[\underline{f}^8, \bar{f}^8] = [\underline{\mu}_{\tilde{F}^8_1}(3) \times \underline{\mu}_{\tilde{F}^8_2}(6), \bar{\mu}_{\tilde{F}^8_1}(3) \times \bar{\mu}_{\tilde{F}^8_2}(6)] = [0.50 \times 0.48, 0.67 \\ \times 0.65] = [0.24, 0.43]\end{aligned} \tag{10.58}$$

$y^{\text{COS}}_{c2}(x_1, x_2)$ is computed in two steps: (1) COS type-reduction, followed by (2) defuzzification.

(1) COS *type-reduction*: $Y^{\text{cos}}_{c2}(3, 6)$ is computed as:

$$Y^{\text{cos}}_{c2}(3,6) = \int_{c_2 \in [\underline{C}^2_{\text{avg}}, \bar{C}^2_{\text{avg}}]} \cdots \int_{c_{14} = \in [\underline{C}^{14}_{\text{avg}}, \bar{C}^{14}_{\text{avg}}]} \int_{f_2 \in [\underline{f}_2, \bar{f}_2]} \cdots \int_{f_{14} \in [\underline{f}_{14}, \bar{f}_{14}]} 1 \Big/ \frac{\sum_{i \in \Omega} c_i f_i(3,6)}{\sum_{i \in \Omega} f_i(3,6)} \; \Omega = \{2, 3, 4, 7, 8, 9, 12, 13, 14\} \tag{10.59}$$

Table 10.14 Firing intervals needed to compute $y_{c2}^{\text{COS}}(x_1, x_2)$

Fired rule (i)	$\mu_{\tilde{F}_1^i}(3)$	$\mu_{\tilde{F}_2^i}(6)$	Firing interval $[\underline{f}^l, \bar{f}^l]$
2	[0, 0.25]	[0, 0.38]	[0, 0.09]
3	[0, 0.25]	[0.48, 0.65]	[0, 0.16]
4	[0, 0.25]	[0, 0.28]	[0, 0.07]
7	[0.50, 0.67]	[0, 0.38]	[0, 0.26]
8	[0.50, 0.67]	[0.48, 0.65]	[0.24, 0.43]
9	[0.50, 0.67]	[0, 0.28]	[0, 0.19]
12	[0, 0.25]	[0, 0.38]	[0, 0.10]
13	[0, 0.25]	[0.48, 0.65]	[0, 0.16]
14	[0, 0.25]	[0, 0.28]	[0, 0.07]

Note that (to within two significant figures) the primary membership intervals associated with $\mu_{\tilde{F}_1^l}(3)$ and $\mu_{\tilde{F}_2^l}(6)$ can be read directly off of Fig. 10.6a

In (10.59) $c_i \in [\underline{C}_{\text{avg}}^i, \bar{C}_{\text{avg}}^i]$ are given in the last column of Table 10.13, and $f_i \in [\underline{f}_i, \bar{f}_i]$ are given in Table 10.14. EIASC or EKM algorithms are used to compute $Y_{c2}^{\text{COS}}(3,6)$ as $[3.51, 5.88]$.

(2) *Defuzzification*: $y_{c2}^{\text{COS}}(3,6) = (3.51 + 5.88)/2 = 4.70$.

Comparing $y_{c2}^{\text{COS}}(x_1, x_2)$ with its type-1 counterpart in (4.60), $y_{c1}(3,6) = 4.85$, observe that: (a) $y_{c1}(3,6) \in Y_{c2}^{\text{COS}}(3,6)$, and (b) $y_{c1}(3,6) \neq y_{c2}^{\text{COS}}(3,6)$. The first observation is what one would have expected, but it is reassuring to see that it actually occurs. The second observation means that the type-reduced set, $Y_{c2}^{\text{COS}}(3,6)$, is not symmetrically situated about $y_{c1}(3,6)$. Uncertainties do not necessarily flow in a symmetrical manner about the output of the IT2 FLA.

Plots of $y_{c2}^{\cos}(x_1)$, $y_{c2}^{\cos}(x_2)$, and $y_{c2}^{\cos}(x_1, x_2)$ are depicted in Figs. 10.7, 10.8 and 10.9, respectively, for $\rho = 0.5$. The dashed lines on these plots are the defuzzified outputs of the comparable Sect. 4.4.5 T1 FLA, namely, $y_{c1}(x_1)$, $y_{c1}(x_2)$ and $y_{c1}(x_1, x_2)$. Complete 3D plots of $y_{c2,l}(x_1, x_2)$ and $y_{c2,r}(x_1, x_2)$ are not shown on the same graph, because they obscure each other; hence, cross-sections of these plots are shown in Fig. 10.9. All of these plots reveal that the IT2 FLAs provide a banded output, so that these FLAs can be used as described in Sect. 4.4.8.

Plots of $y_{c2}^{\cos}(x_1)$ and $y_{c2}^{\cos}(x_2)$ for $\rho = 1$ are depicted in Figs. 10.10 and 10.11, respectively. Comparing these plots with their respective plots in Figs. 10.7 and 10.8, observe that:

- The increased uncertainty in the FOUs have indeed propagated all the way through each FLA to its output; i.e., the bands in Figs. 10.10 and 10.11 are larger than those in Figs. 10.7 and 10.8.
- The bands in Figs. 10.10 and 10.11 are severely non-monotonic, something that is undesirable. The regions of non-monotonicity seem to occur in the regions where there is a large overlap among the FOUs (see Fig. 10.6b). Such large

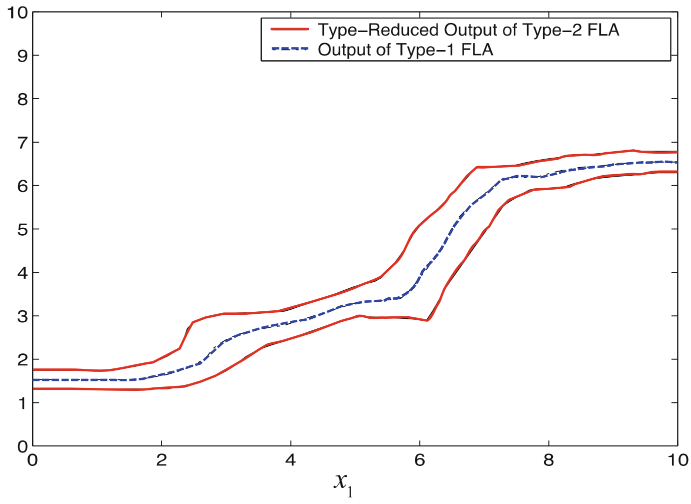


Fig. 10.7 Outputs of the (50%) IT2 and T1 FLAs for the indicator x_1

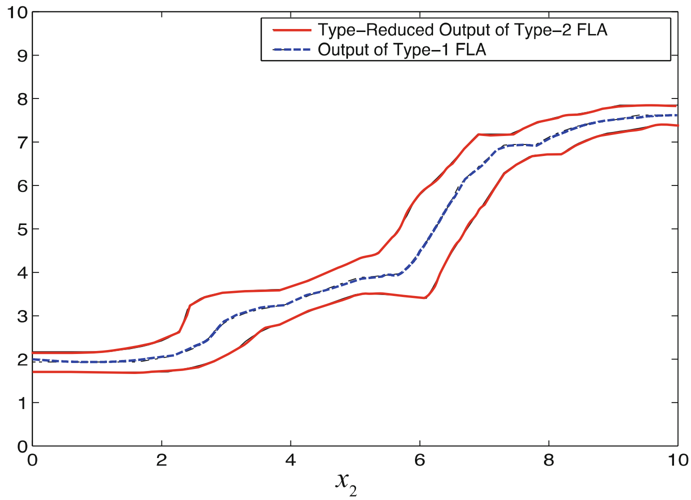


Fig. 10.8 Outputs of the (50%) IT2 and T1 FLAs for the indicator x_2

overlap does not occur for the $\rho = 0.5$ FOUs (see Fig. 10.6a). In retrospect, a 100% uncertainty level is too large; for, if one was that uncertain then one should have chosen the T1 MFs (the dashed curves in Fig. 10.6) to be associated with the $\pm 2\sigma$ points instead of with the $\pm\sigma$ points as has been done here.

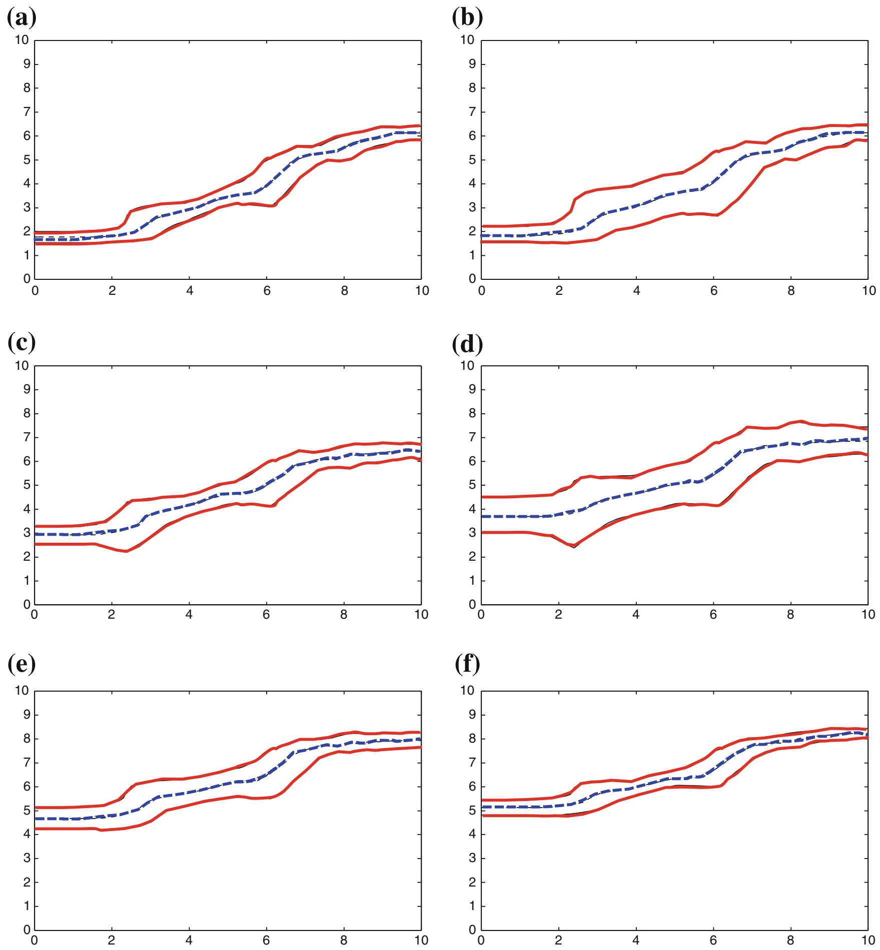


Fig. 10.9 Outputs of the (50%) IT2 and T1 FLAs for the indicators x_1 and x_2. **a** $x_1 = 0$, **b** $x_1 = 2$, **c** $x_1 = 4$, **d** $x_1 = 6$, **e** $x_1 = 8$, and **f** $x_1 = 10$. The *horizontal axis* of each plot is x_2

Using an IT2 FLA it is now possible to propagate the different kinds of uncertainties that occur in survey-based knowledge mining, from the inputs of the FLA to its output, something that could not be done using a T1 FLA.

10.4.4 Remark

Sections 4.4.6 ("Preserving all of the responses") and 4.4.7 ("On multiple indicators") apply as is to IT2 FLAs, and so they are not repeated here.

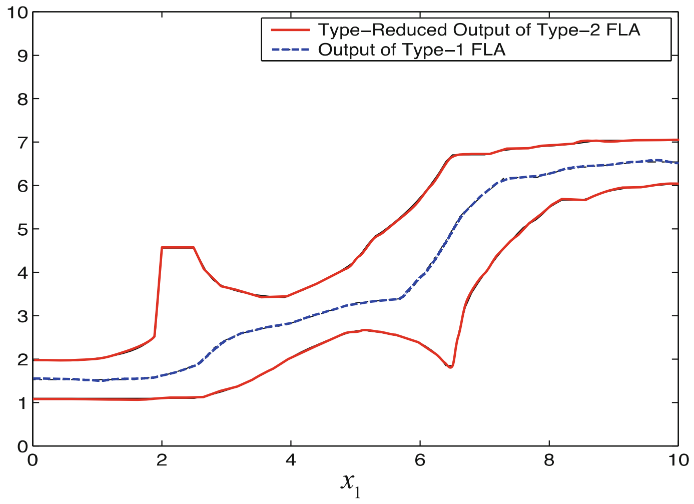


Fig. 10.10 Outputs of the (100%) IT2 and T1 FLAs for the indicator x_1

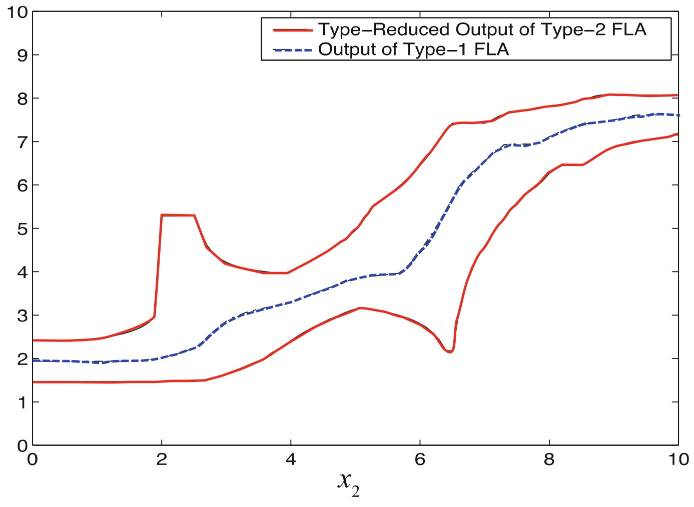


Fig. 10.11 Outputs of the (100%) IT2 and T1 FLAs for the indicator x_2

10.4.5 How to Use the IT2 FLA

As for a T1 FLA, each IT2 FLA that has been designed above can be referred to as a *consensus* IT2 FLA, because it is obtained by using survey results from a population of people.

Recall that Fig. 4.13 depicts one way to use a FLA (T1 or IT2) to advise an individual about a social judgment. It assumes that an individual is given the same questionnaire that was used in Step 6 of the knowledge mining process, which led to the consensus IT2 FLA. Their completed questionnaire can be interpreted as the individual's FLA, and its output can be plotted on the same plot as the output of the consensus FLA. These outputs can then be compared, and if some or all of the individual's outputs are "far" from those of the consensus FLA, then some action could be taken to sensitize the individual about these differences. Figure 4.14 depicts this for a type-1 consensus FLA.

It was pointed out in Sect. 4.4.8 that there is a problem with the type-1 comparisons, namely, how "far" must the differences be between the individual FLA and the consensus FLA before some action (e.g., sensitivity training) is taken? This can be difficult to establish when one is comparing two functions, especially since "far" is in itself a fuzzy term.

This problem is handled directly with the type-2 comparisons in Fig. 10.12. Note that the individual's FLA is still type-1, and has not changed from Figs. 4.14 to 10.12. It is treated as type-1 because the individual takes the survey only one time; hence, there is no uncertainty associated with his or her consequents. The consensus IT2 FLA is represented on Fig. 10.12 by two curves, $y_{c2,l}^{\mathrm{COS}}(x)$ and $y_{c2,r}^{\mathrm{COS}}(x)$. These represent the left-hand (lower) and right-hand (upper) curves, respectively, for the COS type-reduced sets of the consensus IT2 FLA. The difference between these curves represents a measure of the uncertainties due to the words used in the surveys as well as the consensus consequents. Observe from Fig. 10.12 that the individual's FLA curve falls within the bounds of the type-reduced set; hence, no actions need to be taken. This conclusion is quite different from the one that might have been reached by examining the curves in Fig. 4.14 where it appears that there is a significant difference between the individual's behavior level and the consensus T1 FLA's behavior level for larger values of x.

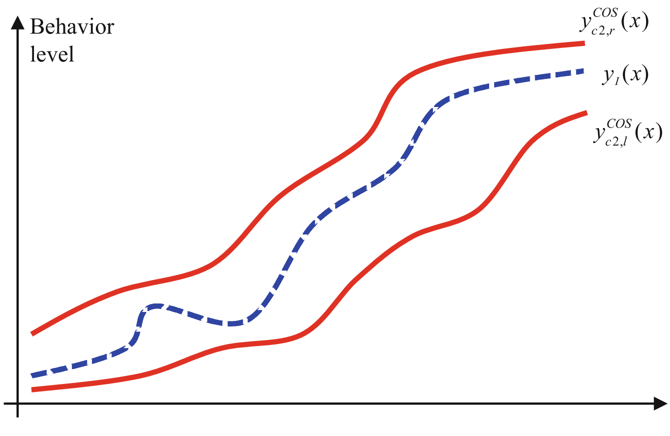


Fig. 10.12 Comparison of consensus IT2 FLA behavior level and an individual's FLA behavior

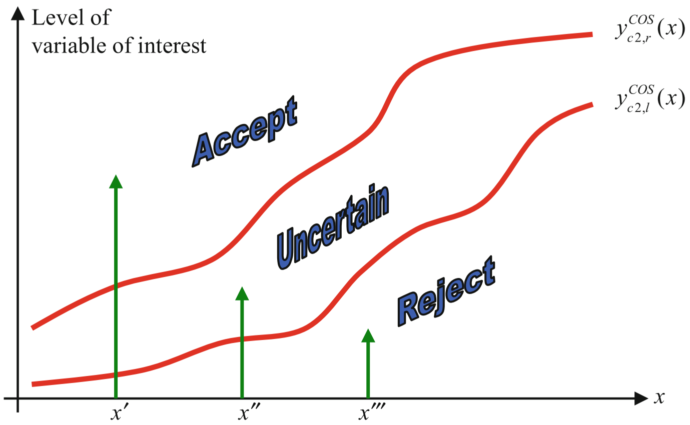


Fig. 10.13 A way to use the IT2 FLA for an engineering judgment. The judgments are to accept when $x = x'$, reject when $x = x'''$, and obtain further clarification when $x = x''$

Recall, also that another way to use an IT2 FLA is depicted in Fig. 4.15. After the consensus FLA has been designed, it is exposed to a situation, say $x = x'$ (in this discussion $\mathbf{x}$ is assumed to be a scalar indicator, x), for which it provides the consensus output $y_{c2}(x')$. Then some action or decision occurs. The problem that was associated with the T1 FLA for a social judgment is the same for an engineering judgment, namely, one would have to take an action or make a decision based only on a point value. This is again resolved by using a consensus IT2 FLA, as depicted in Fig. 10.13. Now the region defined by the type-reduced set (i.e., the "uncertain" region) is one where the designer is free to make a decision. For any $y_{c2,l}^{\mathrm{COS}}(x) < f(x) < y_{c2,r}^{\mathrm{COS}}(x)$, using a different decision boundary will lead to a different engineering judgment. For example, in a petroleum-drilling situation, at $x = x'$ it is clear that one should drill; at $x = x'''$ it is clear that one should not drill; but at $x = x''$ it is not clear what one should do. If $y_{c2,l}^{\mathrm{COS}}(x)$ is used to make the drilling decision, then one drills, but, if $y_{c2,r}^{\mathrm{COS}}(x)$ is used to make the drilling decision, then one does not drill. More information is needed when $x = x''$.

This kind of soft decision has the potential to be used in network control, signal detection and classification, communication receivers, airport security, facial recognition, etc.

10.4.6 Connections to the Perceptual Computer

This would be a good time for the reader to reread Sect. 4.4.9, "Connections to the Perceptual Computer." Flirtation is one example of a social/subjective judgment. Others are described in Mendel and Wu (2010), e.g., assisting in making investment choices, assisting in hierarchical decision making (procurement judgment), and assisting in hierarchical and distributed decision-making (journal publication decision-making). That book extends everything that has been described in Sects. 10.4.1–10.4.3 in the following ways.

Chapter 3 of Mendel and Wu (2010) provides an encoding approach (the *Interval Approach*) for mapping data that are collected from a group of subjects into word FOUs, where the data determines whether the word will be modeled either as a left-shoulder, interior or right-shoulder FOU, which is very different from what is done in Sect. 10.4.1 where the choice of the FOU is made independent of the data. When data are collected from a group of n subjects, they are asked a question like[10] (Hao and Mendel 2015; Liu and Mendel 2008; Mendel and Wu 2010; Wu et al. 2012):

> Suppose that a word can be located on a scale of l to r, and you want to locate the endpoints of the interval that you associate with the word on that scale. On the scale of l to r, what are the endpoints that you associate with the left [right] end-point of the word?

This kind of survey has been administered (by this author) many times and most people have no trouble in answering this question. For each word, the ith subject provides interval endpoints $a^{(i)}$ and $b^{(i)}$, and the group of n subjects provides $[a^{(i)}, b^{(i)}]_{i=1}^{n}$.

Over close to a decade three encoding approaches—*Interval Approach* (IA), *Enhanced Interval Approach* (EIA) and *Hao-Mendel Approach* (HMA)—were developed for mapping such data into an FOU.[11] All three approaches use IT2 FSs as the word model because the MF of a T1 FS does not provide flexibility for *simultaneously* incorporating the intra- and inter-uncertainties of a word (see Sect. 5.2).

Mendel (2003, 2007c) explains that it is scientifically incorrect to model a word using a T1 FS,[12] and Mendel and John (2002) and Mendel (2007b) explain that an IT2 FS is a first-order uncertainty model for a word, because all of its secondary grades are the same. These IT2 FS word models can be thought of as *first-order word granules* (Bargiela and Pedrcyz 2003).

The IA (Liu and Mendel 2008) was replaced by the EIA (Wu et al. 2012), which was subsequently replaced by the HMA (Hao and Mendel 2015). The EIA makes use of more information in the group of n data intervals than does the IA, and the HMA makes use of even more information in the group of n data intervals than

[10]See, also, Sect. 5.2.

[11]Another approach for obtaining an IT2 FS word model from the same kind of data that are used by the IA, EIA and HMA is in Tahayori and Sadeghian (2012). Its Median Interval Approach (MIA) is based on calculating the median boundaries of the range of MFs associated with the words.

[12]A T1 FS can be used either to model each subject's intra-uncertainty, or to model the inter-uncertainty of a group of subjects (by using group statistics), but it cannot model both kinds of uncertainties simultaneously.

does the EIA. An interesting feature of the HMA is that the word FOUs are completely normal (i.e., both their UMF and LMF are normal T1 FSs), whereas only the UMFs from the IA and EIA are normal T1 FSs.

Regarding the Engine of the Perceptual Computer, Mendel and Wu (2010, Chaps. 5 and 6) describe two kinds of engines—if-then rules and novel weighted averages. For if-then rules, they advocate determining a firing level rather than a firing interval, by using the Jaccard similarity measure for IT2 FSs (Exercise 7.46), so that the final combined IT2 FS is more similar looking to an application's codebook FOU than is the final combined IT2 FS obtained when firing intervals are used.

Novel weighted averages range from the IWA (Sect. 8.2) to the fuzzy weighted average (which only uses T1 FSs—see Exercise 8.15) to the linguistic weighted average (which uses IT2 FSs, or a mixture of T1 and IT2 FSs). The latter is a weighted average, where weights and evaluations are linguistic terms, whose FOUs can be modeled, e.g. by using the HMA. Another very powerful NWA is the linguistic weighted power mean (Rickard et al. 2011, 2013).

Regarding the Decoder of the Perceptual Computer (Mendel and Wu 2010, Chap. 4), similarity and subsethood (Exercise 7.47) play very important roles.

An important aspect of the Perceptual Computer is that the complete vocabulary of all of the words that are used in an application must be established before IT2 FS models are found for the words. The size of the vocabulary for a linguistic variable affects the calibration of the fuzzy sets. If, for example, only three linguistic terms are used to describe Profitable, namely {*hardly profitable, moderately profitable, fully profitable*}, then their fuzzy sets will look very different from their fuzzy sets when the following six terms are used: {*barely profitable, hardly profitable, somewhat profitable, moderately profitable, fully profitable, extremely profitable*}. This is because the term *barely profitable* now appears before *hardly profitable*, and the term *extremely profitable* now appears after *fully profitable*. So, knowing the complete vocabulary for all of the linguistic variables is crucial to the proper modeling of the words in an application.

Another interesting aspect of the Perceptual Computer is that it can only be interacted with using words that are in the codebook. When words are modeled as IT2 FSs, and, e.g., the Engine is if-then rules, then one is always in the situation of IT2 non-singleton fuzzification! That is the bad news. The good news is that since the vocabulary and codebook are known ahead of time, all possible firing intervals can be pre-computed and then stored in a look-up table.

Finally, Chap. 6 in Mendel and Wu (2010) gives all of the details for a *Perceptual Computer Flirtation Advisor*, using the same data that have been used in this book.

10.5 Forecasting of Compressed Video Traffic Using IT2 Mamdani and TSK Fuzzy Systems

This problem is fully explained in Sect. 4.5 and is to forecast I frame sizes (i.e., the number of bits/frame[13]) for *Jurassic Park*, using IT2 TSK and Mamdani fuzzy systems. In this section the following two designs of fuzzy system forecasters are examined, based on the logarithm of the first 1000 I frame sizes of *Jurassic Park*, $s(1)$, $s(2)$, …, $s(1000)$ (see Fig. 4.17): singleton normalized A2-C0 IT2 TSK fuzzy system and singleton IT2 Mamdani fuzzy system that uses height type-reduction + defuzzification. The first 504 data [$s(1)$, $s(2)$, …, $s(504)$] were used for tuning the parameters of these forecasters, and the remaining 496 data [$s(505)$, $s(502)$, …, $s(1000)$] were used for testing after tuning.

Singleton normalized A2-C0 IT2 TSK fuzzy system: The rules of this fuzzy system forecaster are $(l = 1, \ldots, M)$

$$\begin{aligned} \tilde{R}^l_{TSK} :\ & \text{IF}\, s(k-3)\, \text{is}\, \tilde{F}^l_1\, \text{and}\, s(k-2)\, \text{is}\, \tilde{F}^l_2\, \text{and}\, s(k-1)\, \text{is}\, \tilde{F}^l_3\, \text{and}\, s(k)\, \text{is}\, \tilde{F}^l_4, \\ & \text{THEN}\ \hat{s}^l(k+1) = c^l_0 + c^l_1 s(k-3) + c^l_2 s(k-2) + c^l_3 s(k-1) + c^l_4 s(k) \end{aligned} \tag{10.60}$$

In (10.60), $\tilde{F}^l_j$ were chosen initially to be the same for all l and j, and Gaussian primary MFs were used for them, ones with a fixed mean and an uncertain standard deviation. As was done for the singleton type-1 TSK fuzzy system (see Sect. 4.5.1), the initial mean was chosen from the first 500 I frames as $m = 4.73$; however, in this case, for which $\sigma \in [\sigma_1, \sigma_2]$, the 500 I frames were divided into five 100-frame segments from which the initial values of σ^i_1 and σ^i_2 $(i = 1, \ldots, M)$ were computed, using the standard deviations for the jth segment, σ^j, as

$$\sigma^i_1 = \min_{j=1,2,\ldots,5} \sigma^j = 0.06 \tag{10.61}$$

$$\sigma^i_2 = \max_{j=1,2,\ldots,5} \sigma^j = 0.11 \tag{10.62}$$

The number of design parameters for this IT2 TSK fuzzy system is $(4p+1)M = 17M$. $\hat{s}^l(k+1)$ was computed by using (10.1) and the FBFs for this kind of a TSK fuzzy system (Exercise 9.20) in which $y_{TSK}(\mathbf{x}) \equiv \hat{s}^l(k+1)$, $x_1 \equiv s(k-3)$, $x_2 \equiv s(k-2)$, $x_3 \equiv s(k-1)$ and $x_4 \equiv s(k)$.

Singleton IT2 Mamdani fuzzy system that uses height type-reduction + defuzzification: The rules of this fuzzy system forecaster are $(l = 1, \ldots, M)$

[13]For those who are note interested in the specifics of this application, just view Fig. 4.17 as a time series, where "time" is the same as "frame index."

$$\begin{aligned}\tilde{R}_Z^l:\ &\text{IF } s(k-3) \text{ is } \tilde{F}_1^l \text{ and } s(k-2) \text{ is } \tilde{F}_2^l \text{ and } s(k-1) \text{ is } \tilde{F}_3^l \text{ and } s(k) \text{ is } \tilde{F}_4^l,\\ &\text{THEN } \hat{s}^l(k+1) = \tilde{G}^l\end{aligned} \tag{10.63}$$

Height type-reduction was used, and $\tilde{F}_j^l$ were chosen initially as just described for the singleton normalized A2-C0 IT2 TSK fuzzy system. The same kind of Gaussian primary MF was used for $\tilde{G}^i$, i.e., one with a fixed mean and an uncertain standard deviation. However, because height type-reduction was used, the consequent MFs are only characterized by $\bar{y}^i$ instead of by interval endpoints, as would have been the case if COS type-reduction had been used. The number of design parameters for this IT2 Mamdani fuzzy system is $(3p+1)M = 13M$. $\hat{s}^l(k+1)$ was computed by using (10.1) and the FBFs for this kind of a IT2 Mamdani fuzzy system (Exercise 9.19) in which $y_h(\mathbf{x}) \equiv \hat{s}^l(k+1)$, $x_1 \equiv s(k-3)$, $x_2 \equiv s(k-2)$, $x_3 \equiv s(k-1)$ and $x_4 \equiv s(k)$.

10.5.1 Forecasting I Frame Sizes: Using the Same Number of Rules

In this first approach to designing the IT2 fuzzy system forecasters, the number of rules was fixed at five, i.e., $M = 5$. Doing this means that the singleton normalized A2-C0 IT2 TSK fuzzy system is described by 85 design parameters, and the singleton IT2 Mamdani fuzzy system that uses height type-reduction + defuzzification is described by 65 design parameters. Steepest descent algorithms (as described in Sect. 10.2.3) were used to tune all of these parameters, in which step sizes (learning parameters) of $\beta_\theta = 0.001$ and $\beta_\theta = 0.01$ were used for the IT2 TSK and Mamdani fuzzy systems, respectively.

The Sect. 4.5.2 discussions on the choice of initial values for MF parameters, use of 50 Monte Carlo realizations and 10 epochs of training for each design and, the RMSE that was used to evaluate the designs [see (4.64)], apply here as well.

The average value and standard deviations of the RMSEs are plotted in Fig. 10.14 for each of the 10 epochs not only for the two IT2 fuzzy system forecasters, but also for the Sect. 4.5.2 two T1 fuzzy system forecasters.

Observe, From Fig. 10.14a, that:

1. The IT2 TSK fuzzy system outperforms the type-1 TSK fuzzy system at each epoch.
2. The IT2 Mamdani fuzzy system outperforms the type-1 Mamdani fuzzy system at each epoch.
3. At the end of the first epoch, the IT2 Mamdani fuzzy system has the best performance. It outperforms the IT2 TSK fuzzy system by [(0.112–0.091)/0.112] $\times$ 100 $\cong$ 18.75%, and outperforms the type-1 TSK fuzzy system by [(0.134–0.091)/0.134] $\times$ 100 $\cong$ 32%. This again demonstrates that an IT2 Mamdani fuzzy system is very promising for adaptive signal processing where more than one epoch of tuning is not possible.

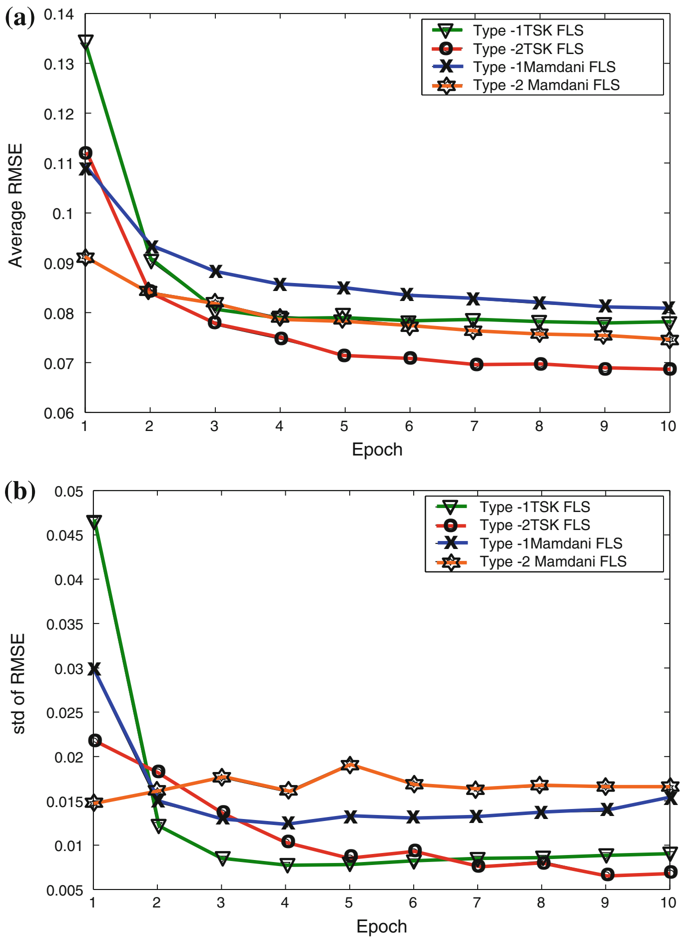


Fig. 10.14 The mean and standard deviations of RMSE (using the test data) for the four five-rule fuzzy system forecasters, averaged over 50 Monte Carlo designs. Tuning was performed in each design for 10 epochs. **a** Mean values, and **b** standard deviations. Note that "FLS" is equivalent to "fuzzy system"

4. After 10 epochs of tuning, the average RMSE of the 4 fuzzy system forecasters is:

   - Type-1 TSK fuzzy system: 0.078
   - IT2 TSK fuzzy system: 0.069
   - Singleton type-1 Mamdani fuzzy system: 0.081
   - Singleton IT2 Mamdani fuzzy system: 0.075

   Observe, from Fig. 10.14b, that:

1. At the end of the first epoch, the IT2 Mamdani fuzzy system has the smallest RMSE standard deviation.
2. After 10 epochs of training, the IT2 TSK fuzzy system has the smallest RMSE standard deviation.

In conclusion, after 10 epochs of tuning the IT2 TSK fuzzy system gives the best performance of all four designs.

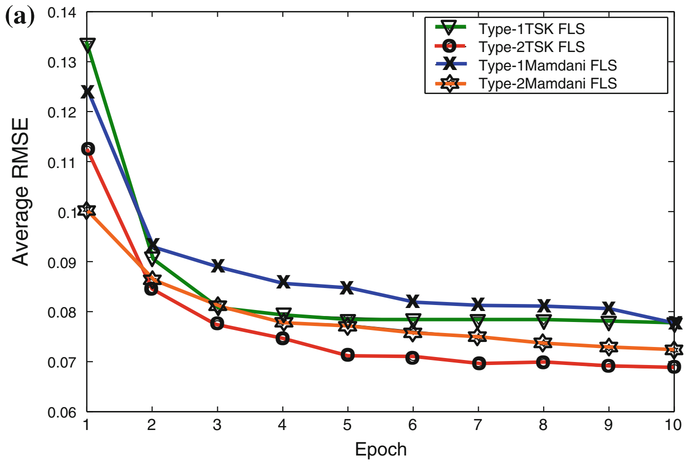
10.5.2 Forecasting I Frame Sizes: Using the Same Number of Design Parameters

Because a five-rule singleton IT2 TSK fuzzy system always has more parameters (design degrees of freedom) to tune than does a comparable five-rule IT2 Mamdani fuzzy system, the previous approach to designing the two fuzzy systems was modified (as it was in Sect. 4.5.3) by fixing the rules used by the singleton IT2 TSK fuzzy system at five and by then choosing the number of rules used by the singleton IT2 Mamdani fuzzy system so that its total number of design parameters approximately equals the number for the IT2 TSK fuzzy system. Doing this led to using seven rules for the singleton IT2 Mamdani fuzzy system. The designs of the resulting two fuzzy systems proceeded exactly as described in the preceding section. All designs were again evaluated using the RMSE in (4.64). The average value and standard deviations of these RMSEs are plotted in Fig. 10.15 for each of the 10 epochs. Observe that:

1. The results are similar to the ones depicted in Fig. 10.14, so, at least for this example, equalizing the numbers of design parameters in the IT2 Mamdani and TSK fuzzy systems does not seem to be so important.
2. The IT2 Mamdani and TSK fuzzy systems still outperform their type-1 counterparts.
3. After only one epoch of tuning the IT2 Mamdani fuzzy system still perform the best.
4. After 10 epochs of tuning the IT2 TSK fuzzy system performs the best.

10.5.3 Conclusion

It is not our intention in this example to recommend one fuzzy system architecture over another. Stepping back from the details of the two simulations, one can see that the IT2 TSK and Mamdani fuzzy systems outperformed their type-1 counterparts, supporting our contention that uncertainties can be better handled by IT2 fuzzy systems.



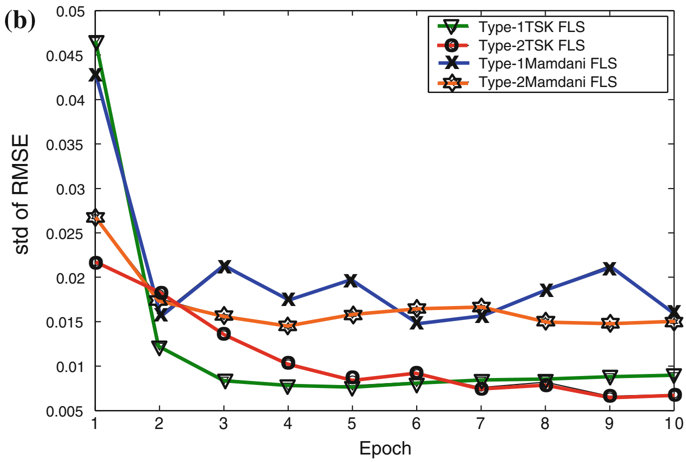


Fig. 10.15 The mean and standard deviations of RMSE (using the test data) for four fuzzy system forecasters, averaged over 50 Monte Carlo designs. The T1 TSK and Mamdani fuzzy systems have approximately the same number of design parameters, as did the IT2 TSK and Mamdani fuzzy systems. Tuning was performed in each design for 10 epochs. **a** Mean values, and **b** standard deviations. Note that “FLS” is the same as “fuzzy system”

10.6 IT2 Rule-Based Classification of Video Traffic

This section continues the exploration of the rule-based approach to a high-level video classification problem that was explained in Sect. 4.6. The problem that is examined is the direct classification of compressed (MPEG-1) video traffic without decompressing it. This is for high-level classification, e.g., classify a video as a movie or a sports program, or as a movie or documentary. As is stated in Sect. 4.6, given a collection of MPEG-1 compressed movies and sports program videos, a subset of them are used to create (i.e., design and test) a rule-based classifier (RBC) in the framework of fuzzy systems. The overall approach taken here is to:

1. Choose appropriate features that act as the antecedents in an IT2 RBC
2. Establish the FOUs for the features
3. Establish rules using the features
4. Optimize the rule design-parameters using a tuning procedure
5. Evaluate the performance of the optimized IT2 RBC using testing

Regarding the features, the logarithm of bits per I, P, and B frame are used as the three features of the IT2 RBC. See Sect. 4.6.1 for further discussions about them.

10.6.1 FOUs for the Features

Although Sect. 4.6.2 stated that "Liang and Mendel (2001) showed that the logarithm of I/P/B frames sizes are more appropriately modeled as Gaussians," in actuality they showed that the logarithm of I, P, or B frame sizes are more appropriately modeled as Gaussians *each of whose mean is a constant, but whose standard deviation varies*. This suggests that one should use an FOU for a Gaussian primary MF with a fixed mean and an uncertain standard deviation (as in Fig. 6.16) to model each frame size of the compressed video. Consequently, in this section the Fig. 6.16 FOU is used for the features in the IT2 RBC.

10.6.2 Rules and Their Parameters

As in Sect. 4.6.3, rules for an IT2 RBC of compressed video traffic use the three selected features as their antecedents and have one consequent. The antecedents are: logarithm of bits/I frame, logarithm of bits/P frame, and logarithm of bits/B frame. The consequent is +1 if the video is a movie and –1 if it is a sports program. There is nothing fuzzy about a rule's consequent in rule-based classification; i.e., each rule's consequent is assigned a numerical value, +1 or −1.

Each rule in an *IT2 RBC* has the following structure:

$$\tilde{R}_Z^l : \text{IF I frame is } \tilde{F}_1^l \text{ and P frame is } \tilde{F}_2^l \text{ and B frame is } \tilde{F}_3^l\text{, THEN the product is a movie } (+1) \text{ or a sports program } (-1) \tag{10.64}$$

This rule is a special case of an IT2 Zadeh rule, one in which the consequent is a singleton, i.e., (10.64) can be expressed as:

$$\tilde{R}_Z^l : \text{IF I frame is } \tilde{F}_1^l \text{ and P frame is } \tilde{F}_2^l \text{ and B frame is } \tilde{F}_3^l, \text{ THEN}$$
$$y^l = \begin{cases} 1 & \text{for a movie} \\ -1 & \text{for a sports program} \end{cases} \tag{10.65}$$

As in Sect. 4.6.3, only one rule per video product was used in this study, e.g., if the training set contains four movies and four sports programs, only eight rules were used.

Each antecedent FOU has three design parameters, its mean and two standard deviation endpoints; hence, there are nine design parameters per rule. Optimum values for all design parameters were determined during a tuning process.

10.6.3 Fuzzifiers

Three IT2 RBCs were designed, one each for the three kind of fuzzifiers that can be used in an IT2 fuzzy system, namely singleton, T1 non-singleton and IT2 non-singleton. For T1 non-singleton fuzzification, a Gaussian MF was used whose mean was located at the measurement's value and whose standard deviation parameter was estimated. This adds three more parameters that have to be tuned, one for each measurement. For IT2 non-singleton fuzzification, the Fig. 6.16 FOU was used whose mean was located at the measurement's value and whose two standard deviation parameters were estimated. This adds six more parameters that have to be tuned, two for each measurement.

10.6.4 Computational Formulas for the IT2 RBCs

It has already been mentioned that the rules in (10.64) are a special case of an IT2 Zadeh rule, one in which the consequent is a singleton. Rule consequent, y^l, is treated as a crisp set; i.e., $y^l = 1$ for a movie, and $y^l = -1$ for a sports program. The MF $\mu_{G^l}(y)$ for this crisp set is $(l = 1, \ldots, M)$

$$\mu_{G^l}(y) = \begin{cases} 1 & y = y^l \\ 0 & \text{otherwise} \end{cases} \tag{10.66}$$

To illustrate the computational formulas for an IT2 RBC, attention here is directed at the IT2 non-singleton fuzzifier case (Exercise 10.18 asks the reader to obtain the comparable formulas for the singleton and T1 non-singleton fuzzifier cases). Because, it is the sign of the defuzzified output of the IT2 RBC that is the basis for classification (see Sect. 4.6.4), our choice for the architecture of the

IT2 RBC was the unnormalized A2-C0 IT2 TSK fuzzy system, in which its consequents are constants.[14] From Sect. 9.6.4.1, it follows that:

$$y^U_{\mathrm{RBC},2}(\mathbf{x}') = \frac{1}{2}[y^U_{\mathrm{TSK},l}(\mathbf{x}') + y^U_{\mathrm{TSK},r}(\mathbf{x}')] \tag{10.67}$$

$$y^U_{\mathrm{TSK},l}(\mathbf{x}') = \sum_{i=1}^{M} c_0^i \underline{f}^i(\mathbf{x}') = \sum_{i=1}^{M} y^i \underline{f}^i(\mathbf{x}') \tag{10.68}$$

$$y^U_{\mathrm{TSK},r}(\mathbf{x}') = \sum_{i=1}^{M} c_0^i \bar{f}^i(\mathbf{x}') = \sum_{i=1}^{M} y^i \bar{f}^i(\mathbf{x}') \tag{10.69}$$

in which $y^i = \pm 1$, so that:

$$y^U_{\mathrm{RBC},2}(\mathbf{x}') = \frac{1}{2}\sum_{i=1}^{M} y^i \left[\underline{f}^i(\mathbf{x}') + \underline{f}^i(\mathbf{x}')\right] \tag{10.70}$$

In (10.69), $\underline{f}^i(\mathbf{x}')$ and $\underline{f}^i(\mathbf{x}')$ are computed by using (9.82) and (9.83) for IT2 non-singleton fuzzification. The final decision that the measurements correspond to either a movie or a sports program is based on the sign of the defuzzified output, i.e.,

$$\begin{array}{ll} \text{IF } y^U_{\mathrm{RBC},2}(\mathbf{x}') > 0 & \text{decide movie} \\ \text{IF } y^U_{\mathrm{RBC},2}(\mathbf{x}') < 0 & \text{decide sports program} \end{array} \tag{10.71}$$

If $y^U_{\mathrm{RBC},2}(\mathbf{x}') = 0$, flip a fair coin to decide whether the measurements correspond to a movie or a sports program.

10.6.5 Optimization of the Rule Design Parameters

The simulation results to be discussed in Sect. 10.6.6 begin with five movies and five sports programs (as they do in Sect. 4.6.7) and, by way of illustration, are for IT2 RBCs that use four movie rules and four sports program rules; i.e., each classifier has eight rules. Each IT2 RBC is optimized by modifying the Sect. 10.2.3 steepest descent tuning procedure, for the A2-C0 unnormalized IT2 TSK fuzzy system, to the situation where $c_j^l = 0$ for $j = 1, \ldots, p$.

---

[14]Earlier it was argued that such a TSK fuzzy system is better referred to as a Mamdani fuzzy system; however, because there is no unnormalized Mamdani fuzzy system, but there is an unnormalized TSK fuzzy system, the designation "TSK" is used here.

10.6.6 Results and Conclusions

As in Sect. 4.6.7, here the focus is on one set of results for so-called *out-of-product classification*. "Out-of-product" means that *some* of the compressed data are used from *some* of the *available* video products to establish the rules and to optimize (tune) the resulting classifier, after which the classifier is tested on the unused video products. This is explained further in Sect. 4.6.7. Average false alarm rate (FAR; see footnote 24 in Chap. 4) is the performance measure that is used to evaluate the three IT2 RBCs and to compare them against a Bayesian classifier (Sect. 4.6.6) and a singleton type-1 RBC (Sect. 4.6.4). The results are:

- Singleton type-1 fuzzy RBC: FAR = 9.41%
- Singleton IT2 RBC: FAR = 13.65%
- T1 non-singleton IT2 RBC: FAR = 8.43%
- IT2 non-singleton IT2 RBC: FAR = 8.03%
- Bayesian classifier: FAR = 14.29%

From these results, observe that the IT2 non-singleton IT2 RBC provides the best performance, and has 44% fewer false alarms than does the Bayesian classifier. This is because the IT2 non-singleton IT2 RBC allows for variations in the standard deviations of I, P, and B frame sizes whereas the Bayesian classifier does not. Additional simulation studies that use 20 video products (10 movies and 10 sports programs) have been performed and support these conclusions.

In summary, it has been demonstrated that it is indeed possible to perform high-level classification of movies and sports programs working directly with compressed data. This seems to be possible because an IT2 RBC allows the uncertainties about the variable standard deviations of the logarithm of the frame sizes to be handled within the framework of IT2 FSs. These uncertainties are not only accounted for by the FOUs of the rule antecedents, but are also accounted for by modeling the measurements as IT2 fuzzy numbers.

Another application of rule-based classification using real data, in which results are obtained and compared for non-singleton T1 and IT2 non-singleton IT2 RBCs and a Bayesian classifier, is Wu and Mendel (2007). This is a multi-category classification problem of battlefield ground vehicles using acoustic features, in which one must distinguish between heavy-tracked, light-tracked, heavy-wheeled, and light-wheeled vehicles.

10.7 Equalization of Time-Varying Nonlinear Digital Communication Channels

When a message (e.g., speech) gets confused because of the transmitting and receiving media (e.g., telephone system) as well as by objects that may interfere with it (e.g., tall buildings, mountains), one says that there is intersymbol

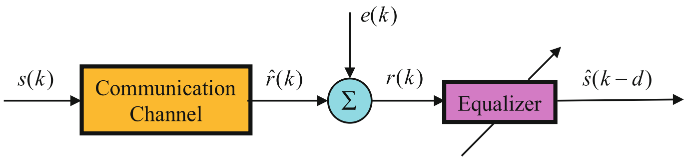


Fig. 10.16 Block diagram of a baseband communication system that is subject to ISI and AGN (Karnik et al. 1999; ©1999, IEEE)

interference (ISI). For the message to be understood at the receiving end, inter-symbol interference must be undone. This is accomplished at the receiving end by hardware and software and is known as *equalization*.[15]

The block diagram for a baseband communication system subject to ISI[16] and additive Gaussian noise (AGN) is depicted in Fig. 10.16, where $s(k)$ is the symbol to be transmitted, $e(k)$ is the AGN, and the measured channel output $r(k)$ can be represented, e.g., for a linear channel, as:

$$r(k) = \hat{r}(k) + e(k) = \sum_{i=0}^{n} a_i(k)s(k-i) + e(k) \tag{10.72}$$

In (10.72), the channel order is n (i.e., there are n + 1 taps), time-varying tap coefficients are $a_i(k)$ $(i = 1, 2, \ldots, n)$, and $s(k)$ is assumed to be binary, either +1 or –1 with equal probability. The goal in channel equalization is to recover the input sequence $s(k)$ $(k = 1, 2, \ldots)$ based on a sequence of $r(k)$ values without knowing or estimating the channel coefficients. This is accomplished with the equalizer block, whose output can be expressed as

$$\hat{s}(k-d) = \mathrm{sign}(f(\text{window of past measurements})) = \begin{cases} +1 & \text{If } f(\cdot) \geq 0 \\ -1 & \text{If } f(\cdot) < 0 \end{cases} \tag{10.73}$$

where d is a decision delay, and $f(\cdot)$ denotes some non-linear operation on a window of past measurements. Usually, $f(\cdot)$ is referred to as the *equalizer*.

According to Wikipedia,[17] an *adaptive equalizer* is an equalizer that automatically adapts to time-varying properties of the communication channel, mitigating

[15]The material in this section is taken from Karnik et al. (1999) and Liang and Mendel(2000b).

[16]Definitions of the terms used in this section can be found in any standard textbook on communication theory, e.g. Proakis (1989).

[17]Wikipedia/Adaptive equalizer: accessed on June 27, 2016.

the effects of multipath propagation and Doppler spreading.[18] In today's communication environment (e.g., mobile communications), the channels are time varying, and the classical equalizers do not perform well for rapidly time-varying channels.

The time-varying nature of a channel is interpreted here as uncertainties in its coefficients. Such uncertainties motivated the use of an IT2 fuzzy system as an adaptive equalizer for time-varying channels. Such an equalizer is referred to here as a *fuzzy adaptive filter* (FAF).

Before the structure of the FAF can be provided, some preliminaries and additional motivation for why IT2 FAFs are more appropriate then T1 FAFs need to be provided.

10.7.1 Preliminaries for Channel Equalization

A very popular architecture for the equalizer $f(\cdot)$ is that of a *transversal equalizer*, for which the window of past measurements is $r(k)$, $r(k-1)$, …, $r(k-p+1)$, where p is the order of the equalizer (i.e., the number of its taps). In this section discussions are limited to transversal equalizers. Let

$$\mathbf{r}(k) \equiv [r(k), r(k-1), \ldots, r(k-p+1)]^T \tag{10.74}$$

Observe, from (10.72) and (10.74), that $\mathbf{r}(k)$ depends on the channel input sequence $s(k), s(k-1), \ldots, s(k-n-p+1)$, which is collected into the following $(n+p) \times 1$ vector

$$\mathbf{s}(k) = [s(k), s(k-1), \ldots, s(k-n-p+1)]^{\mathrm{T}} \tag{10.75}$$

Because $s(k)$ can be +1 or –1, there are $n_s = 2^{n+p}$ possible combinations of the channel input sequence.

In Fig. 10.16, the noise-free signal is $\hat{r}(k)$, where, e.g., for a linear channel:

$$\hat{r}(k) = \sum_{i=0}^{n} a_i(k) s(k-i) \tag{10.76}$$

Let

$$\hat{\mathbf{r}}(k) \equiv [\hat{r}(k), \hat{r}(k-1), \ldots, \hat{r}(k-p+1)]^{\mathrm{T}} \tag{10.77}$$

[18] Work that has been done in the area of adaptive equalization, but mainly for time-invariant channels, includes Proakis (1989) and the many references therein, and, Chen et al. (1993a, b, 1995), Cowan and Semnani (1998), Lee (1996), Moon and Jeon (1998), Patra and Mulgrew (1998), Sarwal and Srinath (1995), Savazzi et al. (1998), and Wang and Mendel (1993).

where $\hat{\mathbf{r}}(k)$ is called the *channel state* (Chen et al. 1993b). Observe, from (10.77) and (10.76), that each of the $n_s = 2^{n+p}$ combinations of the channel input sequence $\mathbf{s}(k)$ generates one $\hat{\mathbf{r}}(k)$, which is denoted as $\hat{\mathbf{r}}_i(k)$, where $(i = 1, \ldots, n_s)$

$$\hat{\mathbf{r}}_i(k) \equiv [\hat{r}_i(k), \hat{r}_i(k-1), \ldots, \hat{r}_i(k-p+1)]^{\mathrm{T}} \tag{10.78}$$

Hence, each channel state has a probability of occurrence equal to $1/n_s$.

A correct decision by the equalizer occurs if

$$\hat{s}(k-d) = s(k-d) \tag{10.79}$$

Based on the category of $s(k-d)$ (i.e., $\pm$ 1), the channel states $\hat{\mathbf{r}}(k)$ can be partitioned into two classes Chen et al. (1993b)

$$R^{+} = \{\hat{\mathbf{r}}(k) | s(k-d) = 1\} \tag{10.80}$$

$$R^{-} = \{\hat{\mathbf{r}}(k) | s(k-d) = -1\} \tag{10.81}$$

The number of elements in R^{+} and R^{-} are denoted n_s^{+} and n_s^{-}, respectively. Because $s(k-d)$ has equal probability to be +1 or –1, it follows that $n_s^{+} = n_s^{-} = 2^{n+p-1}$. The channel states in R^{+} and R^{-} are denoted $\hat{\mathbf{r}}_i^{+}$ $(i = 1, \ldots, n_s^{+})$ and $\hat{\mathbf{r}}_i^{-}$ $(i = 1, \ldots, n_s^{-})$, respectively.

Example 10.12 Consider the following time-invariant *nonlinear* channel model that was used in Wang and Mendel (1993):

$$r(k) = a_1 s(k) + a_2 s(k-1) - 0.9[a_1 s(k) + a_2 s(k-1)]^3 + e(k) \tag{10.82}$$

where $a_1 = 1$ and $a_2 = 0.5$, as shown in Fig. 10.17a. Here it is assumed that the decision delay $d = 0$. From Table 10.15, observe that there are eight channel states and that $s(k)$ determines which category (+1 or -1) $\hat{\mathbf{r}}(k)$ belongs to. Observe also that $[\hat{r}(k), \hat{r}(k-1)]$ in the first four rows of Table 10.15 have category +1 [as determined by (10.80) for $s(k-d) = s(k) = 1$], and $[\hat{r}(k), \hat{r}(k-1)]$ in the last four rows of Table 10.15 have category -1 [as determined by (10.81) for $s(k-d) = s(k) = -1$]. The channel states are plotted in Fig. 10.17b when $a_1 = 1$ and $a_2 = 0.5$, and are simple points. To construct that figure, it was assumed that no additive noise was present, so that $\hat{r}(k) = r(k)$ and $\hat{r}(k-1) = r(k-1)$.

10.7.2 Why an IT2 FAF Is Needed

When a channel's coefficients are time-varying, as illustrated next, the channel's states are no longer simple points, but are instead clusters.

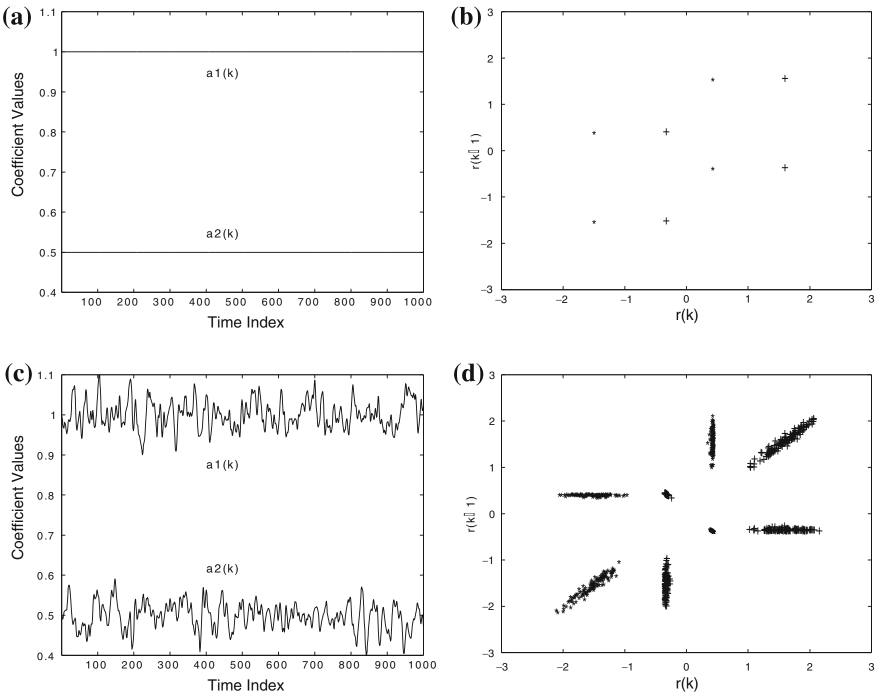


Fig. 10.17 For the channels in (10.82) and (10.83): **a** time-invariant channel coefficients, $a_1 = 1$ and $a_2 = 0.5$; **b** channel states (noise-free) of time-invariant channel, where *asterisk* denotes the category $\hat{\mathbf{r}}(k) = +1$ and *plus* denotes the category $\hat{\mathbf{r}}(k) = -1$; **c** an example of time-varying channel coefficients when $\beta = 0.1$; and **d** channel states (noise-free) of the time-varying channel whose coefficients are the ones in **c** (Karnik et al. 1999; ©1999, IEEE), and (Liang and Mendel 2000b; ©2000, IEEE)

Example 10.13 Here the results of Example 10.12 are generalized to the time-varying version of the nonlinear channel in (10.82), i.e. to

$$r(k) = a_1(k)s(k) + a_2(k)s(k-1) - 0.9[a_1(k)s(k) + a_2(k)s(k-1)]^3 + e(k) \tag{10.83}$$

where a_1 and a_2 are time-varying coefficients, each simulated, as in Cowan and Semnani (1998), by using a second-order Markov model in which a white Gaussian noise source drives a second-order Butterworth low-pass filter. Note that $a_1(k)$ has been centered about 1, $a_2(k)$ has been centered about 0.5, and the white Gaussian input to the Butterworth filter has a standard deviation equal to β.

Realizations of the time-varying coefficients and channel states are plotted in Fig. 10.17c, d, respectively, for a moderate noise level of $\beta = 0.1$. The results in (d) are based on $\hat{r}(k)$ and $\hat{r}(k-1)$ given in Table 10.15. Observe that the channel

Table 10.15 Channel states for the nonlinear channel model in (10.82) with binary symbols: $d = 0$ and $p = 2$ (Karnik et al. 1999) and (Liang and Mendel 2000b)

			Channel states	
$s(k)$	$s(k-1)$	$s(k-2)$	$\hat{r}(k)$	$\hat{r}(k-1)$
1	1	1	$a_1(k) + a_2(k) - 0.9[a_1(k) + a_2(k)]^3$	$a_1(k) + a_2(k) - 0.9[a_1(k) + a_2(k)]^3$
1	1	−1	$a_1(k) + a_2(k) - 0.9[a_1(k) + a_2(k)]^3$	$a_1(k) - a_2(k) - 0.9[a_1(k) - a_2(k)]^3$
1	−1	1	$a_1(k) - a_2(k) - 0.9[a_1(k) - a_2(k)]^3$	$-a_1(k) + a_2(k) - 0.9[-a_1(k) + a_2(k)]^3$
1	−1	−1	$a_1(k) - a_2(k) - 0.9[a_1(k) - a_2(k)]^3$	$-a_1(k) - a_2(k) - 0.9[-a_1(k) - a_2(k)]^3$
−1	1	1	$-a_1(k) + a_2(k) - 0.9[-a_1(k) + a_2(k)]^3$	$a_1(k) + a_2(k) - 0.9[a_1(k) + a_2(k)]^3$
−1	1	−1	$-a_1(k) + a_2(k) - 0.9[-a_1(k) + a_2(k)]^3$	$a_1(k) - a_2(k) - 0.9[a_1(k) - a_2(k)]^3$
−1	−1	1	$-a_1(k) - a_2(k) - 0.9[-a_1(k) - a_2(k)]^3$	$-a_1(k) + a_2(k) - 0.9[-a_1(k) + a_2(k)]^3$
−1	−1	−1	$-a_1(k) - a_2(k) - 0.9[-a_1(k) - a_2(k)]^3$	$-a_1(k) - a_2(k) - 0.9[-a_1(k) - a_2(k)]^3$

states are now eight clusters instead of eight individual points. These clusters illustrate that $\hat{\mathbf{r}}_i$ is uncertain for all $i = 1, \ldots, 8$ when the coefficients of the channel are time-varying.

For a T1 FAF, type-1 Gaussian MFs were chosen for rule antecedents that are centered at the single-channel states for rule antecedents. For an IT2 FAF, type-2 MFs were chosen, namely Gaussian primary MFs with uncertain means, given in Example 6.17, where $\sigma_k^l \equiv \sigma_e$. The latter choice was motivated by the projections of the channel state clusters onto each of their two axes. By understanding the nature of the channel states when a channel's coefficients are varying one is again led to a specific FOU.

10.7.3 Designing the IT2 FAFs

Here the designs of singleton T1 and singleton IT2 FAFs are illustrated for the non-linear time-varying channel in (10.83). Each FAF has eight TSK rules (in which the consequents are constants[19]), one per channel state. The rules in a *singleton T1 FAF* have the following structure $(l = 1, \ldots, 8)$:

$$R^l_{\mathrm{TSK}} : \text{IF}\, r(k)\, \text{is}\, F_1^l \;\text{and}\; r(k-1)\, \text{is}\, F_2^l,\ \text{THEN}\, y^l = w_l \tag{10.84}$$

They have the following structure in a *singleton IT2 FAF* $(l = 1, \ldots, 8)$:

$$\tilde{R}^l_{\mathrm{TSK}} :\ \text{IF}\, r(k)\, \text{is}\, \tilde{F}_1^l \;\text{and}\; r(k-1)\, \text{is}\, \tilde{F}_2^l,\ \text{THEN}\, y^l = w_l \tag{10.85}$$

In these rules, w_l is a crisp value of +1 or –1, as determined by:

$$w_i \equiv \begin{cases} +1 & \hat{\mathbf{r}}(k) \in R^+ \\ -1 & \hat{\mathbf{r}}(k) \in R^- \end{cases} \tag{10.86}$$

For the T1 rules, as just mentioned, Gaussian MFs are used for F_1^l and F_2^l; and, for the IT2 rules, the Gaussian primary MFs in Example 6.17 are used for $\tilde{F}_1^l$ and $\tilde{F}_2^l$, where $\sigma_k^l \equiv \sigma_e$. As also just explained, the range of the mean of the primary MF for antecedent $\tilde{F}_1^l$ ($\tilde{F}_2^l$) corresponds to the horizontal (vertical) projection of the lth cluster depicted in Fig. 10.17d.

Because of the isomorphism between equalization and classification, the computational formulas for T1 and IT2 FAF's are easily obtained from Sects. 4.6.4 and 10.6.4, respectively, and are not restated here.

In Karnik et al. (1999) and Liang and Mendel (2000b), the mean-value parameters of all MFs were estimated using a clustering procedure that was applied

[19]See footnote 14.

to some training data, as in (Chen et al. 1993a), because such a procedure is computationally simple. This same procedure was used here (an alternative to doing this is to use tuning).

To complete the specification of the MFs, a value was also needed for the standard deviation, σ_e, of the AGN $e(k)$. Chen et al. (1995) showed that equalizer performance is not very sensitive to the value of σ_e; hence, in the following simulations it was assumed that the value of σ_e is known exactly. Of course, σ_e could also have been estimated during a tuning process using some training data.

10.7.4 Simulations and Conclusions

Here, singleton T1 and singleton IT2 FAFs are compared with a K-nearest neighbor classifier (NNC) (Savazzi et al. 1998) for equalization of the time-varying nonlinear channel in (10.83). Comparable results for a different channel can be found in Mendel (2000). In the simulations, the number of taps of the equalizer, p, was chosen equal to the number of taps of the channel, $n+1$, where $n=1$; i.e. $p=n+1=2$. The number of rules equaled the number of clusters; i.e., $2^{p+n}=8$. A sequence $s(k)$ of length 1000 was used for the experiments. The first 121 symbols[20] were used for training (i.e., clustering) and the remaining 879 were used for testing. The training sequence established the parameters of the antecedent MFs, as described in Sect. 10.7.3. After training, the parameters of the T1 and IT2 FAFs were fixed and then testing was performed.

In the first experiment, SNR was fixed at 20 dB and simulations were run for eight different values of noise level β, ranging from $\beta=0.04$ to $\beta=0.32$ (at equal increments of 0.04), and d was set equal to 0. 100 Monte Carlo simulations were performed for each value of β, where in each realization the channel coefficients and the AGN were uncertain. The mean values and standard deviations of the bit error rate (BER) for the 100 Monte Carlo realizations are plotted in Fig. 10.18a, b, respectively.

In the second experiment, β was fixed at 0.1 and simulations were run for five different SNR values, ranging from SNR = 15dB to SNR = 25dB (at equal increments of 2.5 dB), and again d was set equal to 0. 100 Monte Carlo simulations were performed for each value of SNR, where in each realization the channel coefficients and the AGN were uncertain. The mean values and standard deviations of the BER for the 100 Monte Carlo realizations are plotted in Fig. 10.19a, b, respectively. Observe, from Figs. 10.18 and 10.19, that:

- In terms of the mean values of BER, the IT2 FAF performs better than both the T1 FAF and NNC (see Figs. 10.18a and 10.19a).

[20]In the K-NNC, if the number of training prototypes is N, then $K=\sqrt{N}$ is the optimal choice for K. It is required that N be an odd integer; hence, the choice of $N=121$.

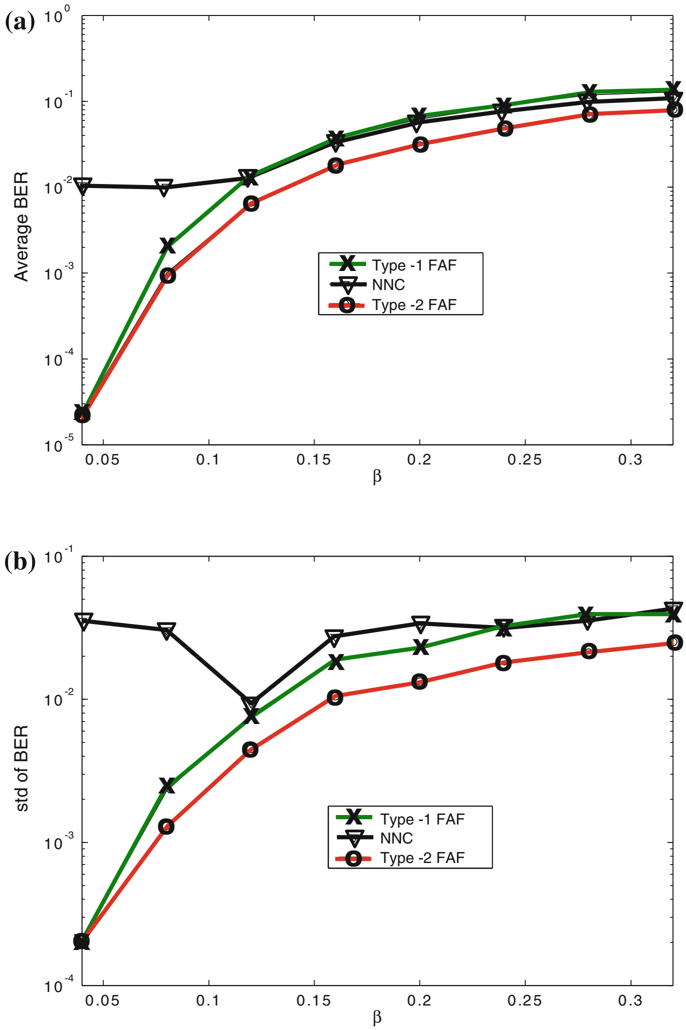


Fig. 10.18 Performance of T1 FAF, NNC, and IT2 FAF versus β when SNR = 20dB and the number of training prototypes is 121. **a** Average BER, and **b** standard deviation (std) of BER for 100 Monte Carlo realizations (Liang and Mendel 2000b; © 2000 IEEE)

- When SNR = 20 dB and $\beta \geq 0.12$, the NNC performs about the same as the T1 FAF, but the T1 FAF performs better than the NNC when $\beta < 0.12$. However, regardless of β, the IT2 FAF always performs better than the NNC (see Fig. 10.18a).
- In terms of the standard deviation of BER, the IT2 FAF is more robust to the AGN than are the other two equalizers; and, the T1 FAF is also more robust than the NNC (see Figs. 10.18b and 10.19b].

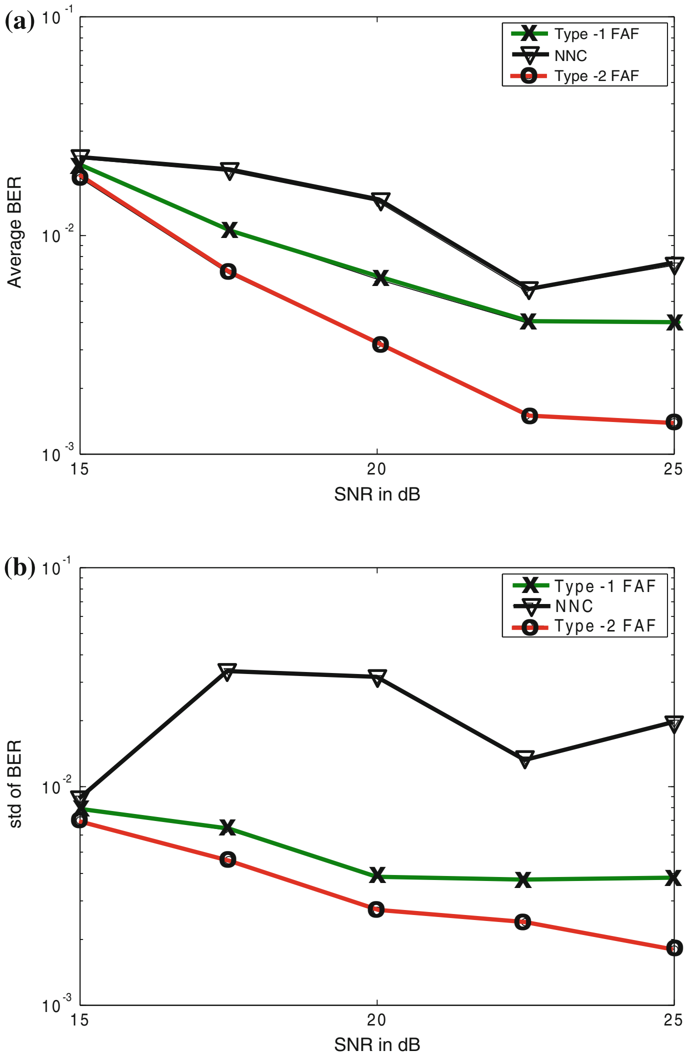


Fig. 10.19 Performance of T1 FAF, NNC, and IT2 FAF versus SNR when $\beta = 0.1$ and the number of training prototypes is 121. **a** Average BER, and **b** standard deviation (std) of BER for 100 Monte Carlo realizations

These observations suggest that an IT2 FAF, as just designed, looks very promising as a good transversal equalizer for time-varying nonlinear channels. This is because an IT2 FAF allows uncertainties about the uncertain nature of the channel states to be modeled within the framework of IT2 FSs. As in the previous RBC application, these uncertainties can be accounted for by the FOUs of the rule antecedents. Our conjecture is that even better performance could have been obtained had the measurements been modeled using IT2 FSs, as was done for the IT2 RBC.

10.8 IT2 Fuzzy Logic Control

Recall that Sect. 4.7 provided a short early history of fuzzy control, explained what a type-1 fuzzy logic controller is, and provided: a short introduction to fuzzy PID control, the general structure of a fuzzy PID controller, conventional and fuzzy PID control design methods, and some simulation results comparing T1 fuzzy PID control with PID control. This section explains what an IT2 fuzzy logic controller is and provides some simulation results comparing a number of IT2 fuzzy PID controls with type-1 fuzzy PID control and PID control.

10.8.1 What Is an IT2 Fuzzy Logic Controller (FLC)?[21]

The IT2 fuzzy system in Fig. 9.1 can be used as an IT2 FLC. The defuzzified output of such an IT2 FLC can be used as the command to an actuator in the control system.

Section 4.7.2 presented four sources of uncertainties that face real-world control systems (including fuzzy logic control systems). In addition to those uncertainties, IT2 FLCs are also affected by:

- Linguistic uncertainties, because the meaning of words that are used in the antecedent's and consequent's linguistic labels can be uncertain, i.e., words mean different things to different FLC designers (Mendel 2001).
- In addition, experts do not always agree and they often provide different consequents for the same antecedents. A survey of experts will usually lead to a histogram of possibilities for the consequent of a rule; this histogram represents the uncertainty about the consequent of a rule (Mendel 2001).

In an IT2 FLC all of these uncertainties are modeled by the FOUs of the antecedents and/or consequents of the rules, as well as by the kind of fuzzifier.

Just as a T1 FLC is a variable structure controller so is an IT2 FLC, and just as a T1 FLC has two architectures, Mamdani and TSK, an IT2 FLC also has those two architectures.

Chapters 4–6 in Mendel et al. (2014) demonstrate that:

1. Using IT2 FSs to represent the FLC inputs and outputs can lead to a smaller FLC rule base because MF uncertainties represented by the FOUs of IT2 FSs let the IT2 MFs cover the same range as T1 FSs, but with a smaller number of terms. This *rule reduction* (at the expense of more complicated MFs) increases as the number of FLC inputs increases.

[21] Some of the material in this section is taken from Mendel et al. (2014, Chap. 1).

Table 10.16 The rule base of the IT2 FPID controller

$E \backslash \Delta E$	$\tilde{N}$	$\tilde{Z}$	$\tilde{P}$
$\tilde{N}$	$\tilde{R}_Z^1 : U = NB = -1$	$\tilde{R}_Z^2 : U = NM = -0.5$	$\tilde{R}_Z^3 : U = Z = 0$
$\tilde{Z}$	$\tilde{R}_Z^4 : U = NM = -0.5$	$\tilde{R}_Z^5 : U = Z = 0$	$\tilde{R}_Z^6 : U = PM = 0.5$
$\tilde{P}$	$\tilde{R}_Z^7 : U = Z = 0$	$\tilde{R}_Z^8 : U = PM = 0.5$	$\tilde{R}_Z^9 : U = PM = 1$

2. An IT2 FLC may give a *smoother control surface* than its T1 counterpart, especially in the region around the steady state (for a PI controller this means as both the error and the change of error approach zero).
3. IT2 FLCs *can realize more complex input–output relationships* than T1 FLCs. This has already been demonstrated in Example 9.10 and is due to an IT2 FLC possibly having more first-and second-order rule partitions than a type-1 FLC. It may also be due to the different ways in which firing interval endpoints are used within the first-and second-order rule partitions (i.e., by IT2 novelty rule partitions). This is explained at the end of Sect. 10.8.3.

10.8.2 IT2 Fuzzy PID Control[22]

This is a continuation of Sect. 4.7.3 ("Fuzzy PID Control"), so the reader should review all of that section. The general structure of the IT2 FPID controller is still the one that is depicted in Fig. 4.21. It uses the Table 10.16 symmetrical 3×3 rule base in which $\tilde{N} =$ negative, $\tilde{Z} =$ zero and $\tilde{P} =$ positive, $NB =$ negative big, $NM =$ negative medium, $PM =$ positive medium, $PB =$ positive big and $Z =$ zero. The rule structure of the IT2 FLC is $(l = 1, \ldots, 9)$:

$$\tilde{R}_Z^l : \text{IF } E \text{ is } \tilde{F}_1^l \text{ and } \Delta E \text{ is } \tilde{F}_2^l \text{ THEN } U \text{ is } G^l \tag{10.87}$$

In (10.87), both E and ΔE are described by the three overlapping FOUs that are depicted in Fig. 10.20 (the UMFs of those FOUs are the T1 FSs that are depicted in Fig. 4.22a), and G^l are still the crisp singletons that are shown in Fig. 4.22b and tabulated in Table 10.16.

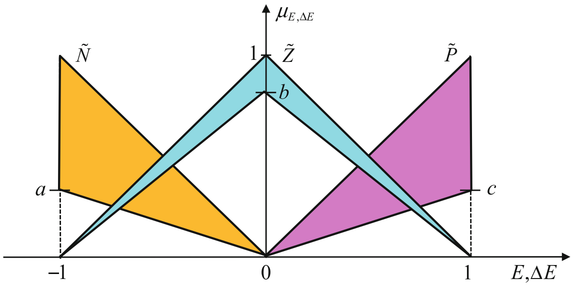
The following four IT2 FPID controllers were implemented that all used singleton fuzzification and product implication:

- Mamdani and center of sets type-reduction[23] + defuzzification (Mamdani: COS TR + D, called "KM" in the figures below)
- WM UB approximate type-reduction + defuzzification (WM UB)

[22]This section was prepared by Prof. Tufan Kumbasar.

[23]KM algorithms were used to compute the COS type-reduced set; however, the same numerical results would have been obtained had the EKM or EAISC algorithms been used.

Fig. 10.20 Illustration of the antecedent FOUs of the IT2 FLC



- NT direct defuzzification (NT)
- BMM direct defuzzification[24] (BMM)

Although different choices are possible for the three FOU parameters (a, b, c) in Fig. 10.20, for this study they were found by tuning all of the FOU parameters for Mamdani: COS TR + D, and are: $a_E = c_E = 0.2$, $b_E = 0.9$, $a_{\Delta E} = c_{\Delta E} = 0.3$, and $b_{\Delta E} = 0.9$. By choosing the parameters in this way, the control surfaces for the FPID controllers are symmetrical.[25]

10.8.3 Simulation Results (IT2-FPID Versus T1-FPID and PID)

This section compares the performances of the internal model control (IMC)-based IT2-FPID, T1-FPID and PID controllers that were designed for the Sect. 4.7.3.3 first-order plus time delay *Nominal Process*, whose parameters are: $K = 1$, $L = 1$ and $T = 1$. Additionally, in order to examine the robustness of the two controllers, the following perturbed processes were considered (the same as in Sect. 4.7.3.4):

- *Perturbed Process-1*: $K = 1.4$, $L = 1.2$ and $T = 1.2$
- *Perturbed Process-2*: $K = 0.6$, $L = 2$ and $T = 0.9$

Step responses for six situations are depicted in Figs. 10.21, 10.22 and 10.23. % overshoot (OS), settling time (T_s) and integral absolute error (IAE) values are given in Table 10.17. The numbers that are in the first two rows of this table were taken from Table 4.15.

As can be clearly seen, the IMC based T1-FPID controller produces superior control performance as compared with the IT2 and conventional PID controllers for the Nominal Process (except for a slightly larger IAE than the conventional PID

[24] In (9.158), $m = n = 1/2$ and y^i equals the crisp consequents that are in Table 10.16.

[25] Better performance may have been obtained for the three other IT2 FPID controllers if the parameters of their FOUs were tuned for each of them.

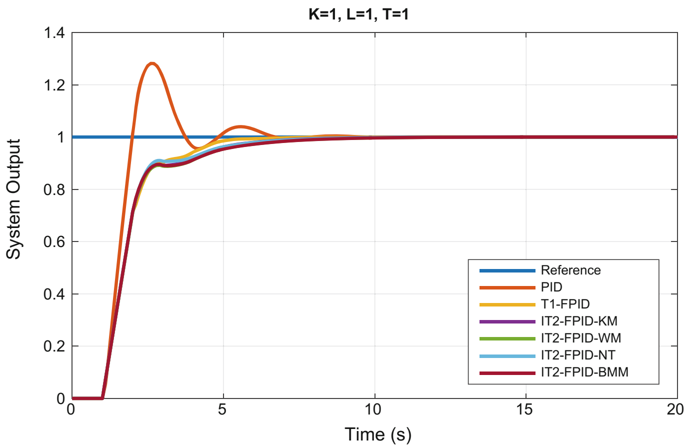


Fig. 10.21 System step responses for the nominal process

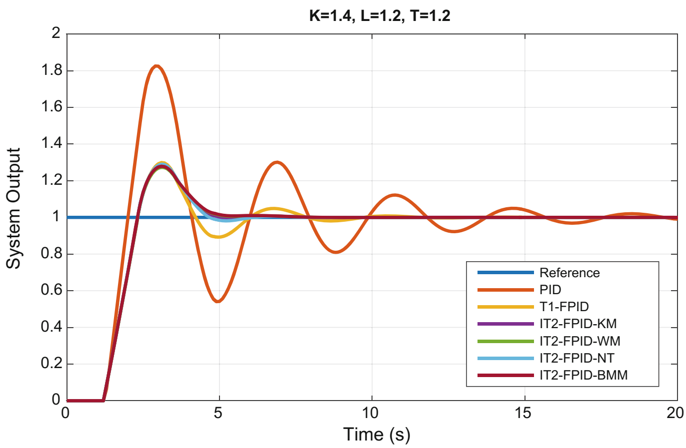


Fig. 10.22 System step responses for the perturbed process-1

controller). It has the same 0% overshoot as do the four IT2 FPID controllers, but its settling time and IAE are smaller than those from all of the IT2 FPID controllers.

It is also very clear from Figs. 10.22 and 10.23 and Table 10.17 that the T1 and IT2 FPID controllers are more robust to changes in the Nominal Process than the conventional PID controller, as is evidenced by their much less oscillatory step response behavior. This robustness is supported by the %OS, T_s and IAE values

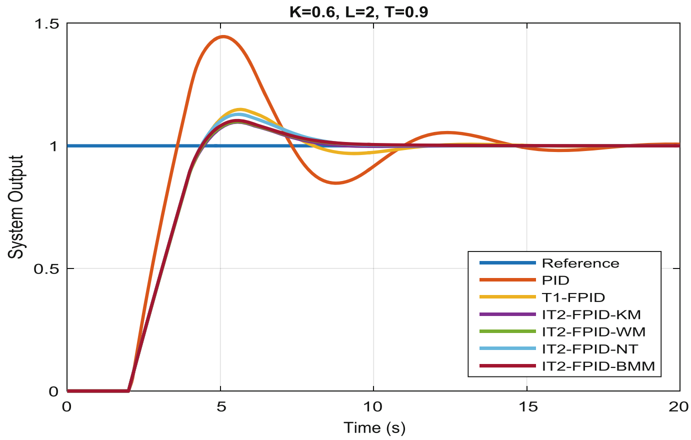


Fig. 10.23 System step responses for the perturbed process-2

that are given in Table 10.17. When these metrics are compared for the T1 and four IT2 FPID controllers, the following conclusions can be drawn[26]:

- For **Perturbed Process-1**, the smallest:
  - % overshoot is achieved by WM UB and all four IT2 FPID controllers outperform the T1 FPID controller
  - Settling time is achieved by KM, and all four IT2 FPID controllers outperform the T1 FPID controller
  - IAE is achieved by KM and NT, and all four IT2 FPID controllers outperform the T1 FPID controller
- For **Perturbed Process-2**, the smallest:
  - % overshoot is achieved by KM and all four IT2 FPID controllers outperform the T1 FPID controller
  - Settling time is achieved by KM, and all of the IT2 FPID controllers outperform the T1 FPID controller
  - IAE is achieved by KM, and all four IT2 FPID controllers outperform the T1 FPID controller
- For **both Perturbed Processes**, the smallest:
  - % overshoot is achieved by WM UB (for Perturbed Process-1) or KM (for Perturbed Process-2)
  - Settling time is achieved by KM

[26]Some of the numbers in this table are very close to each other, and so it is advisable to use statistical tests [e.g., Garcia et al. (2010), Derrac et al. (2011)] to arrive at comparative conclusions. Additionally, in practice, results would be obtained for many more perturbed processes.

Table 10.17 Control performance comparison of the FPID controllers

Controller	Nominal process			Perturbed process-1			Perturbed process-2		
	OS (%)	T_s(s)	IAE	OS (%)	T_s(s)	IAE	OS (%)	T_s(s)	IAE
PID	28.1	6.3	***19.5***	84.3	18.8	43.1	43.6	14.0	43.9
T1-FPID	***0.0***	***4.8***	19.8	29.9	7.6	23.9	14.7	10.5	35.6
IT2-FPID-KM	***0.0***	6.1	20.6	28.3	***4.0***	***22.3***	***9.6***	***7.8***	***33.8***
IT2-FPID-WM UB	***0.0***	6.6	21.0	***26.8***	4.9	22.4	9.8	8.1	34.1
IT2-FPID-NT	***0.0***	6.1	20.4	28.9	4.5	***22.3***	12.8	8.4	34.8
IT2-FPID-BMM	***0.0***	6.6	21.0	27.9	4.9	22.4	10.2	8.2	34.3

Best performance values are in italic

– IAE is achieved by KM (for both perturbed processes) (NT gives the same value but only for Perturbed Process-1)

Control surfaces for the five FPID controllers are given in Fig. 10.24. Comparing the T1, COS TR + D, WM UB, NT, and BMM control surfaces in this figure, a number of observations can be made:

1. The four adjacent endpoints of the surfaces are the same, and it is only the ways in which the adjacent endpoints are connected that are different.
2. The adjacent endpoints of the T1 FLC surface are connected by straight lines.
3. The adjacent endpoints of each of the four IT2 FLC surfaces are connected by curves, each one of which provides a different kind of interpolation between the adjacent endpoints, and so it is in this sense that one may say that *the IT2 FLC surfaces are smoother than the T1 FLC surface*.
4. The IT2 Mamdani FLC (COS TR + D) curves exhibit the most varying curvature, followed by either the IT2 WM UB FLC or the IT2 BMM FLC, whereas the IT2 NT FLC curves are the least varying.
5. The IT2 WM UB, NT, and BMM adjacent end-point curves can also be interpreted as different ways to smooth out the more varying curvature of the IT2 Mamdani FLC (COS TR + D) adjacent end-point curves.
6. The IT2 WM UB and IT2 BMM surfaces look very much alike, which is consistent with the formulaic connections between BMM and WM UB that are given in [Biglarbegian et al. (2010) and in Mendel et al. (2014, Chap. 6)], in which Biglarbegian shows that WM UB can be reorganized so that it equals BMM plus another term. The TSK rules in (10.87) have very simple numerical consequents (namely, constants), which simplifies this analyses. It is conjectured here that, for the present FPID application, the other term is very small.

It is not so easy to discern differences between all of these control surfaces, so, in order to study their differences, Fig. 10.25 depicts the absolute differences between the type-1 and IT2 control surfaces, and Fig. 10.26 depicts the differences between some of the IT2 control surfaces. Examining Fig. 10.25, observe that:

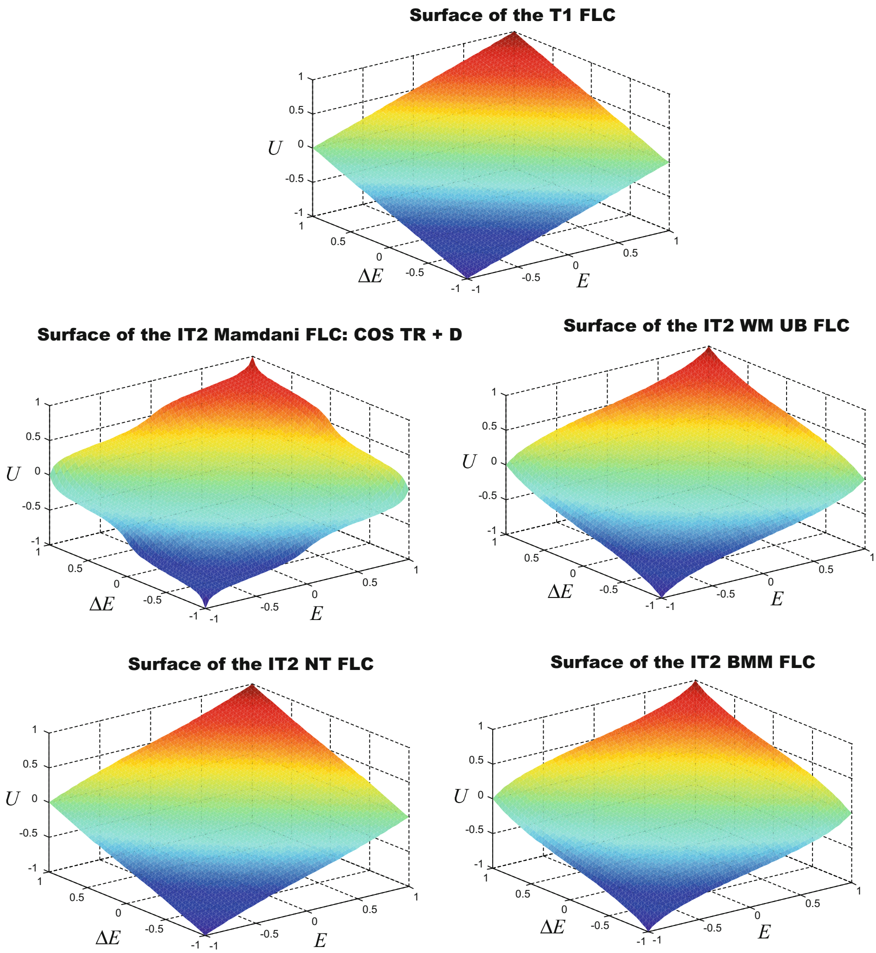


Fig. 10.24 Control surfaces for T1 FLC and the four IT2 FLCs

7. The absolute differences of all of the IT2 FLC surfaces and the T1 FLC surfaces are very noticeable.
8. The absolute differences of the IT2 WM UB and BMM FLC surfaces look very similar, which further supports the statements made above in Item 6.

Examining Fig. 10.26, observe that:

9. The absolute differences of the IT2 Mamdani:(COS TR + D) and WM UB surfaces look very similar to the absolute differences of the IT2 Mamdani: (COS TR + D) and BMM surfaces, which further supports the statements made above in Item 6.

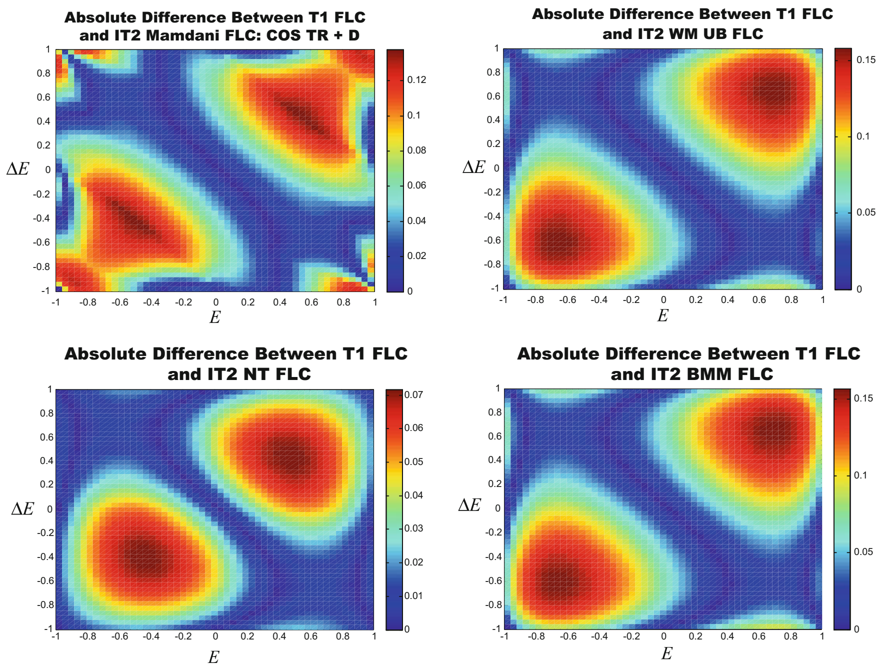


Fig. 10.25 Absolute differences between T1 control surfaces and the IT2 control surfaces

10. The absolute differences of the WM UB and BMM FLC surfaces are not always very small for all values of E and ΔE, so one must be very careful about rushing to hasty conclusions.

In Exercise 10.20, the reader is asked to demonstrate that there are no IT2 second-order rule partitions of $E \times \Delta E$ when E and ΔE are described by the three FOUs that are depicted in Fig. 10.20. And yet, the control surface, in Fig. 10.24, for IT2 Mamdani:(COS type-reduction + D) seems to have much more variability than would be predicted by the absence of IT2 second-order rule partitions.

Recall, that the entire concept of IT2 second-order rule partitions was based on examining how firing intervals are computed within an IT2 first-order rule partition, and since the same firing intervals are computed for all IT2 fuzzy systems, something else must be happening in the IT2 Mamdani: (COS type-reduction + D) FLC that is not happening in the three other IT2 FLCs. Indeed, it is type-reduction that is happening, and during COS type-reduction the endpoints of firing intervals are used in yet another way, a way that has not been accounted for by the concept of an IT2 second-order rule partition, but is accounted for by the concept of an IT2 novelty rule partition (Definition 9.10). The undulating control surface in Fig. 10.24 for COS TR + D depicts the variable nature of these partitions very vividly. Determining formulas for the boundaries of these partitions is very complicated. See Du and Ying (2010), Zhou and Ying (2013), Nie and Tan (2010), and Mendel

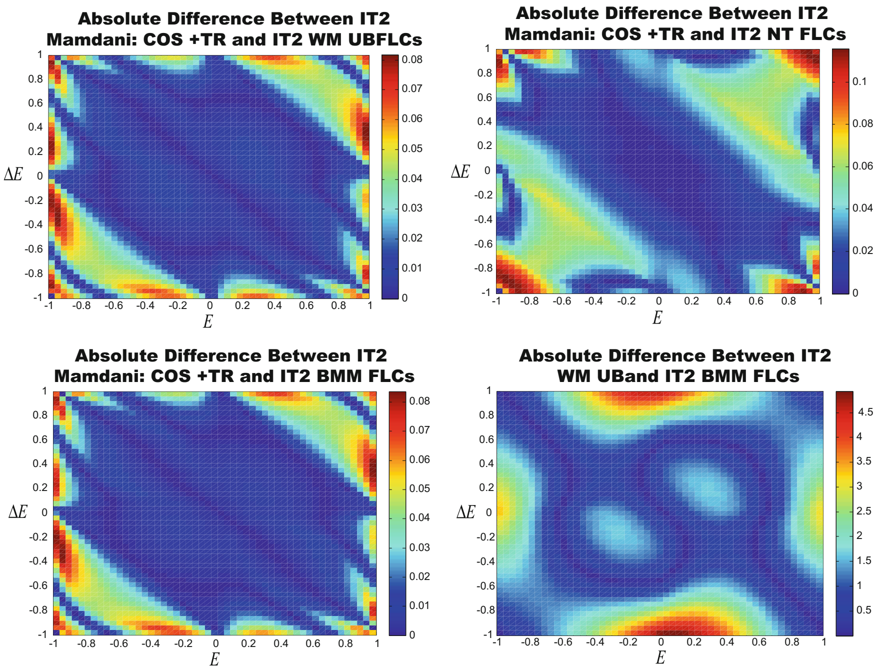


Fig. 10.26 Absolute differences between IT2 control surfaces

et al. (2014, Chaps. 4 and 5) for detailed derivations and pictures of these kinds of partitions[27] for different kinds of IT2 FLCs.

So, after all of these simulations and discussions, which of the IT2 FPID controllers would be the "recommended" one? Without further simulations, KM is the preferred choice, because it is consistently the most robust controller. If, however, one is concerned about the iterative nature of type-reduction, then use NT or BMM.

10.9 Other Applications

When the first edition of this book was written Mendel (2001), almost all applications of IT2 fuzzy systems were by the author's Ph.D. students Nilesh Karnik and Qilian Liang. Since 2001, thousands of articles have been published, many of which are about applications of IT2 fuzzy systems. Readers interested in applications are advised to see Castillo and Melin (2014), Dereli et al. (2011), Hagras (2007), Hagras and Wagner (2012) and Mendel (2007a).

---

[27]In these references the partitions are not referred to as IT2 first-, IT2 second- or IT2 novelty rule partitions. These names and explanations only occurred to this author during the writing of this book.

Exercises

10.1 Suppose that normal isosceles triangles are used for the lower and upper MFs (both share the same apex) of all interior FOUs and parallel piecewise-linear functions are used for the lower and upper MFs for the two shoulder (exterior) FOUs. Redo Examples 10.3–10.6 for these choices.

10.2 Suppose that normal trapezoids are used for the lower and upper MFs (each with a different common overlap) of all interior FOUs and parallel piecewise linear functions are used for the lower and upper MFs for the two shoulder (exterior) FOUs. Redo Examples 10.3–10.6 for these choices.

10.3 (a) Apply the IT2 WM method to the time series and FOUs that are depicted in Fig. 10.27. The IT2 Zadeh rule has five antecedents $x_1^{(t)}$, $x_2^{(t)}, \ldots,$ and $x_5^{(t)}$, and the forecasted value is $x_6^{(t)}$. (b) Compare the IT2 Zadeh rule and its degree with the Example 4.4 Zadeh rule and its degree (e.g., are there any differences?).

10.4 (a) Apply the IT2 WM method to the time series and FOUs that are depicted in Fig. 10.28. Observe that the FOUs in Fig. 10.28 are about twice as fat as those in Fig. 10.27, indicative of much greater uncertainties about the measured values of the time series. The IT2 Zadeh rule has five antecedents $x_1^{(t)}$, $x_2^{(t)}, \ldots,$ and $x_5^{(t)}$, and the forecasted value is $x_6^{(t)}$. (b) Compare the IT2 Zadeh rule and its degree with the Example 4.4 Zadeh rule and its degree (e.g., are there any differences?); (c) Compare the IT2 Zadeh rule and its degree with the Exercise 10.3 Zadeh rule and its degree (e.g., are there any differences?).

10.5 Exercise 3.16 described the truck backing up problem, and Exercise 4.4 provided the data for two initial conditions. Convert Exercise 4.4 into one in which all of the T1 MFs are converted into IT2 FSs. Assume end-point uncertainty for all T1 MFs of $\pm 20\%$ (except for the domain endpoints).

(a) Define the FOUs for all of the terms.
(b) Repeat Part (a) of Exercise 4.4 using the FOUs from Part (a) of this exercise.
(c) Repeat Part (b) of Exercise 4.4 using the FOUs from Part (a) of this exercise.
(d) Repeat Part (c) of Exercise 4.4 using the results from Parts (a) and (b) of this exercise.
(e) What are the differences between the results in Part (d) of this exercise and Part (c) of Exercise 4.4?

10.6 This is a continuation of Exercise 9.19, in which you were asked to obtain the IT2 FBF expansion and the IT2 FBFs for the IT2 Mamdani fuzzy system with height type-reduction + defuzzification. As in Example 10.8,

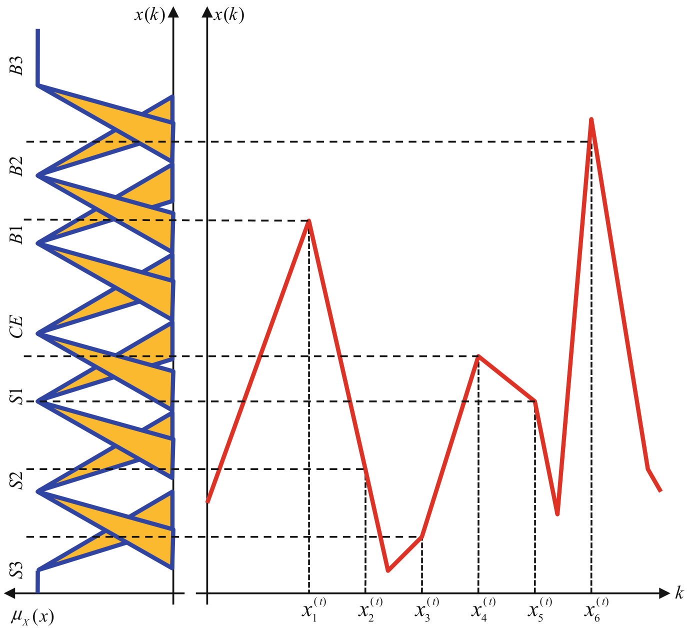


Fig. 10.27 Time series and FOUs for Exercise 10.3

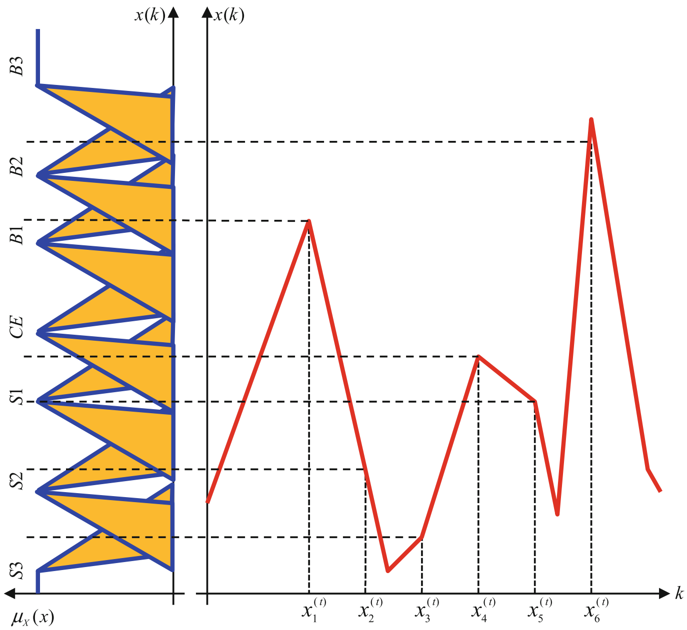


Fig. 10.28 Time series and FOUs for Exercise 10.4

compute the partial derivative of the output of this IT2 fuzzy system with respect to θ_j and the partial derivatives of the IT2 FBFs with respect to θ_j.

10.7 This is a continuation of Exercise 9.20, in which you were asked to obtain the IT2 FBF expansion and the IT2 FBFs for the A2-C0 normalized IT2 TSK fuzzy system. As in Example 10.8, compute the partial derivative of the output of this IT2 fuzzy system with respect to θ_j and the partial derivatives of the IT2 FBFs with respect to θ_j.

10.8 This is a continuation of Exercise 9.21, in which you were asked to obtain the IT2 FBF expansion and the IT2 FBFs for the unnormalized A2-C1 IT2 TSK fuzzy system. As in Example 10.8, compute the partial derivative of the output of this IT2 fuzzy system with respect to θ_j and the partial derivatives of the IT2 FBFs with respect to θ_j.

10.9 This is a continuation of Exercise 9.22, in which you were asked to obtain the IT2 FBF expansion and the IT2 FBFs for the normalized A2-C1 IT2 TSK fuzzy system. As in Example 10.8, compute the partial derivative of the output of this IT2 fuzzy system with respect to θ_j and the partial derivatives of the IT2 FBFs with respect to θ_j.

10.10 Compute the partial derivative of the output of the WM UB output (9.152) with respect to θ_j.

10.11 Compute the partial derivative of the output of the centroid type-reduction + defuzzification IT2 fuzzy system in (9.105) with respect to θ_j.

10.12 Prove that by virtue of the QPSO algorithm, the performance of the optimized singleton IT2 fuzzy system cannot be worse than that of the optimized non-singleton type-1 fuzzy system.

10.13 Prove that by virtue of the QPSO algorithm, the performance of the optimized T1 non-singleton IT2 fuzzy system cannot be worse than that of the optimized singleton IT2 fuzzy system.

10.14 Prove that by virtue of the QPSO algorithm, the performance of the optimized IT2 non-singleton IT2 fuzzy system cannot be worse than that of the optimized T1 non-singleton IT2 fuzzy system.

10.15 Set up the IT2 fuzzy system particles (for each of its three kinds of fuzzifiers) and T1 fuzzy system particles (for each of its two kind of fuzzifiers), as was done in Example 10.10, for the following:

(a) IT2 Mamdani fuzzy system with height type-reduction + defuzzification
(b) Unnormalized A2-C0 IT2 TSK fuzzy system
(c) Normalized A2-C0 IT2 TSK fuzzy system
(d) WM UB IT2 fuzzy system
(e) NT IT2 fuzzy system
(f) BMM IT2 fuzzy system

10.16 Time series come in different guises. For example, the Mackey–Glass time series is totally deterministic but its measured values are random because of the additive measurement noise. Suppose that you are going to forecast a different kind of time series, one whose measurements are perfect, but whose signal is random.

(a) Explain what kind of an IT2 fuzzy system you would choose to do this and why.
(b) What would be a reasonable choice for the FOU?

10.17 In Example 10.11, compute $y_{c2}^{\text{COS}}(x_1, x_2)$ when $(x_1, x_2) = (2, 4)$ and $\rho = 0.5$.

10.18 Section 10.6.4 obtained the computational formulas for an IT2 RBC in the IT2 non-singleton fuzzifier case. They are (10.67)–(10.71). What changes must be made to these formulas for:

(a) Singleton fuzzification?
(b) T1 non-singleton fuzzification?

10.19 Obtain formulas for the four IT2 Mamdani FPID controllers used in Sect. 10.8.2 for the rule base in Table 10.16.

10.20 Demonstrate that there are no IT2 second-order rule partitions for $E \times \Delta E$ when E and ΔE are as in Fig. 10.20.

10.21 Wu and Tan (2006) proposed[28] a simplified IT2 FLC in which T1 FSs are gradually replaced by their IT2 counterparts until the resulting IT2 FLC meets robustness requirements, starting with the FSs that characterize the region around the steady state. Their philosophy is that, since the computational costs will increase significantly when the number of IT2 FSs increases, as few IT2 FSs as possible should be introduced. For a PI-like FLC, the response near steady state is determined mainly by the control surface around $(E, \Delta E) = (0, 0)$, the origin, which is governed by the middle FSs of E and ΔE. Their procedure for designing a simplified IT2 FLC is:

1. Design a baseline T1 FLC through simulation on a nominal model.
2. Change the most important FS to an IT2 FS. For the two inputs of a PI (PID) FLC, ΔE is more susceptible to noises, so the FS corresponding to zero ΔE is the first to be changed to an IT2 FS.
3. If the IT2 FLC designed above cannot cope well with the actual plant, the FS associated with zero E is then also changed to an IT2 FS.
4. If the resulting IT2 FLC is still not robust enough, more IT2 FSs may be introduced starting from the middle of each input domain and

[28]The statement of this exercise is taken or adapted for the most part from Mendel et al. (2014, Chap. 5, pp. 207–208).

gradually moving toward the limits of the domains. Another criterion is to use the IT2 FSs to characterize the operating region that needs a smoother control surface.

As a result of this procedure, during the transient stage, the FLC behaves like a T1 FLC since no IT2 FSs are fired. When the output approaches the set point, IT2 FSs will be fired, so that the plant will then be controlled by an IT2 FLC. Smoother control signals will be generated, which help to eliminate oscillations.

(a) Sketch the T1 and IT2 MFs that are associated with this procedure, using the T1 MFs for both E and ΔE that are given in Fig. 4.22a.

(b) Consider a simplified IT2 FLC that has N^2 rules with crisp consequents $u_i\,(i = 1, \ldots, N^2)$, where N is the number of fuzzy sets for each of the two domains. Suppose the first $M\ (0 < M < N^2)$ rules contain only T1 FSs in their antecedents, and the remaining $N^2 - M$ rules have at least one IT2 FS in their antecedents. There will be M firing levels $(f_i,\ i = 1, \ldots, M)$ and $N^2 - M$ firing intervals $(\tilde{f}_i \equiv [\underline{f}_i, \bar{f}_i],\ i = M+1, M+2 \ldots, N^2)$. Assume the FLC is a normalized TSK fuzzy system, whose output can be expressed as (this is an expressive formula and not a computational formula):

$$y_{\mathrm{TSK}}(E, \Delta E) = \frac{\sum_{i=1}^{M} u_i f_i + \sum_{i=M+1}^{N^2} u_i \tilde{f}_i}{\sum_{i=1}^{M} f_i + \sum_{i=M+1}^{N^2} \tilde{f}_i}$$

Show that $y_{\mathrm{TSK}}(E, \Delta E)$ can be expressed as:

$$y_{\mathrm{TSK}}(E, \Delta E) = \frac{\sum_{i=1}^{M} u_i f_i}{\sum_{i=1}^{M} f_i} + \frac{\sum_{i=M+1}^{N^2+1} u_i' \tilde{f}_i}{\sum_{i=M+1}^{N^2+1} \tilde{f}_i}$$

where

$$u' = \begin{cases} u_i - \frac{\sum_{i=1}^{M} u_i f_i}{\sum_{i=1}^{M} f_i} & i = M+1, M+2, \ldots, N^2 \\ 0 & i = N^2 + 1 \end{cases}$$

and $\tilde{f}_{N^2+1} = \sum_{i=1}^{M} f_i$.

(c) Explain how $y_{\mathrm{TSK}}(E, \Delta E)$ can be computed.

References

Bargiela, A., and W. Pedrycz. 2003. *Granular computing: An introduction*. Dordrecht: Kluwer Academic Publishers.

Cara, A.B., C. Wagner, H. Hagras, H. Pomares, and I. Rojas. 2013. Multiobjective optimization and comparison of nonsingleton type-1 and singleton interval type-2 fuzzy logic systems. *IEEE Transcations on Fuzzy Systems* 21: 459–476.

Castillo, O., and P. Melin. 2014. A review on interval type-2 fuzzy logic applications in intelligent control. *Information Sciences* 279: 615–631.

Chen, S., B. Mulgrew, and S. McLaughlin. 1993a. A clustering technique for digital communications channel equalization using radial basis function network. *IEEE Transaction on Neural Networks* 4: 570–579.

Chen, S., B. Mulgrew, and S. McLaughlin. 1993b. Adaptive Bayesian equalizer with decision feedback. *IEEE Transactions on Signal Processing* 41: 2918–2927.

Chen, S., S. McLaughlin, B. Mulgrew, and P.M. Grant. 1995. Adaptive Bayesian decision feedback equalizer for dispersive mobile radio channels. *IEEE Transactions on Communications* 43: 1937–1956.

Cowan, C.F.N., and S. Semnani. 1998. Time-variant equalization using a novel non-linear adaptive structure. *International Journal of Adaptive Control and Signal Processing* 12: 195–206.

Dereli, T., A. Baykasoglu, K. Altun, A. Durmusoglu, and I. Burkhan Turksen. 2011. Industrial applications of type-2 fuzzy sets and systems: A concise review. *Computers in Industry* 62: 125–137.

Derrac, J., S. Garcia, D. Molina, and F. Herrera. 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1: 3–18.

Du, X., and H. Ying. 2010. Derivation and analysis of the analytical structures of the interval type-2 fuzzy-PI and PD controllers. *IEEE Transactions on Fuzzy Systems* 18: 802–814.

Garcia, S., A. Fernandez, J. Luengo, and F. Herrera. 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences* 180: 2044–2064.

Hagras, H. 2007. Type-2 FLCs: A new generation of fuzzy controllers. *IEEE Computational Intelligence Magazine* 2: 30–43.

Hagras, H., and C. Wagner. 2012. Towards the wide spread use of type-2 fuzzy logic systems in real world applications. *IEEE Computational Intelligence Magazine* 7 (3): 14–24.

Hao, M., and J.M. Mendel. 2015. Encoding words into normal interval type-2 fuzzy sets: HM method, accepted for publication in *IEEE Transactions on Fuzzy Systems*.

Karnik, N.N., J.M. Mendel, and Q. Liang. 1999. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 7: 643–658.

Lee, K.Y. 1996. Complex fuzzy adaptive filters with lms algorithm. *IEEE Transactions on Signal Processing* 44: 424–429.

Liang, Q. and J.M. Mendel. 2000a. Decision feedback equalizer for nonlinear time-varying channels using type-2 fuzzy adaptive filters," In *Proceedings of FUZZ-IEEE'00*, San Antonio, TX, May 2000a.

Liang, Q., and J.M. Mendel. 2000b. Equalization of nonlinear time-varying channels using Type-2 fuzzy adaptive filters. *IEEE Transactions on Fuzzy Systems* 8: 551–563.

Liang, Q., and J.M. Mendel. 2000c. Designing interval type-2 fuzzy logic systems using an SVD–QR method: Rule reduction. *Int'ernational Journal of Intelligent Systems* 15: 939–957.

Liang, Q. and J.M. Mendel. 2001. MPEG VBR video traffic modeling and classification using fuzzy techniques. *IEEE Transactions on Fuzzy Systems*, pp. 183–193, Feb. 2001.

Liu, F., and J.M. Mendel. 2008. Encoding words into interval type-2 fuzzy sets using an *Interval Approach*. *IEEE Transactions on Fuzzy Systems* 16: 1503–1521.

Magdon-Ismail, M., A. Nicholson, and Y. Abu-Mostafa. 1998. Financial markets: Very noisy information processing. *Proceedings of IEEE* 86: 2184–2195.

Marín, L.G., F. Valencia and D. Sáez. 2016. Prediction interval based on type-2 fuzzy systems for wind power generation and loads in microgrid control design. In *Proceedings FUZZ-IEEE 2016*, Vancouver, Canada, July 2016.

Mendel, J.M. 2000. Uncertainty, fuzzy logic, and signal processing. *Signal Processing Journal* 80: 913–933.

Mendel, J.M. 2001. *Introduction to rule-based fuzzy logic systems*. Upper Saddle River, NJ: Prentice-Hall.

Mendel, J.M. 2003. Fuzzy sets for words: A new beginning. In *Proceedings of 2003 IEEE int'ernational conference on fuzzy systems*, St. Louis, MO, pp. 37–42, May 2003.

Mendel, J.M. 2004. Computing derivatives in interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 12: 84–98.

Mendel, J.M. 2007a. Advances in type-2 fuzzy sets and systems. *Information Sciences* 177: 84–110.

Mendel, J.M. 2007b. Computing with words and its relationships with fuzzistics. *Information Sciences* 177: 998–1006.

Mendel, J.M. 2007c. Computing with words: Zadeh, Turing, Popper and Occam. *IEEE Computational Intelligence Magazine* 2: 10–17.

Mendel, J.M., H. Hagras, W.-W. Tan, W.W. Melek, and H. Ying. 2014. *Introduction to type-2 fuzzy logic control*. Hoboken, NJ: John Wiley and IEEE Press.

Mendel, J.M., and R.I. John. 2002. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems* 10: 117–127.

Mendel, J.M., and D. Wu. 2010. *Perceptual computing: Aiding people in making subjective judgments*. Hoboken, NJ: Wiley and IEEE Press.

Mendel, J.M. 2014. General type-2 fuzzy logic systems made simple: A tutorial. *IEEE Transactions on Fuzzy Systems* 22: 1162–1182.

Moon, J., and T. Jeon. 1998. Sequence detection for binary ISI channels using signal space partitioning. *IEEE Transactions on Communications* 46: 891–901.

Nie M., and W.W. Tan. 2010. Derivation of the analytical structure of symmetrical IT2 fuzzy PD and PI controllers. In *Proceedings of 2010 IEEE int'ernationl conference on Fuzzy Systems*, Barcelona, Spain, pp. 1–8, July 2010.

Patra, S.K., and B. Mulgrew. 1998. Efficient architecture for Bayesian equalization using fuzzy filters. *IEEE Transactions Circuits and Systems II: Analog and Digital Signal Processing* 45: 812–820.

Proakis, J.G. 1989. *Digital communications*, 2nd ed. New York: McGraw-Hill.

Rickard, J.T., J. Aisbett, R. Yager and G. Gibbon. 2011. Linguistic weighted power means: Comparison with the linguistic weighted average. In *Proceedings FUZZ-IEEE 2011*, 2011 World Congress on Computational Intelligence, pp. 2185–2192, Taipei, Taiwan.

Rickard, J.T., J. Aisbett, R.R. Yager, and G. Gibbon. 2013. Computing with words using weighted power mean aggregation operators. In *Soft computing: State of the art theory and novel applications*, ed. R.R. Yager, A.M. Abbasov, M.Z. Reformat, and S. Shahbazova, 145–160. New York: Springer.

Rutkowski, L. 2004. *Flexible neuro-fuzzy systems: Structures, learning and performance evaluation*. Boston: Kluwer.

Sarwal, P., and M.D. Srinath. 1995. A fuzzy logic system for channel equalization. *IEEE Transactions Fuzzy Systems* 3: 246–249.

Savazzi, P., L. Favalli, E. Costamagna, and A. Mecocci. 1998. A suboptimal approach to channel equalization based on the nearest neighbor rule. *IEEE Journal Selected Areas in Communications* 16: 1640–1648.

Tahayori, H., and A. Sadeghian. 2012. Median interval approach to model words with interval type-2 fuzzy sets. *International Journal of Advanced Intelligence Paradigms* 4 (3): 313–336.

Wang, L.-X., and J.M. Mendel. 1993. Fuzzy adaptive filters, with application to nonlinear channel equalization. *IEEE Transactions Fuzzy Systems* 1 (3): 161–170.

Wu, D., J.M. Mendel, and S. Coupland. 2012. Enhanced interval approach for encoding words into interval type-2 fuzzy sets and its convergence analysis. *IEEE Transactions on Fuzzy Systems* 20: 499–513.

Wu, D., and W.W. Tan. 2006. Genetic learning and performance evaluation of type-2 fuzzy logic controllers. *Engineering Applications of Artificial Intelligence* 19 (8): 829–841.

Wu, H., and J.M. Mendel. 2007. Classification of battlefield ground vehicles using acoustic features and fuzzy logic rule-based classifiers. *IEEE Transactions on Fuzzy Systems* 15: 56–72.

Zhou, H., and H. Ying. 2013. A method for deriving the analytical structure of a broad class of typical interval type-2 Mamdani fuzzy controllers. *IEEE Transactions on Fuzzy Systems* 21: 447–491.

Zhou, S.-M., J.M. Garibaldi, R.I. John, and F. Chiclana. 2009. On constructing parsimonious type-2 fuzzy logic systems via influential rule selection. *IEEE Transactions on Fuzzy Systems* 17: 654–667.

Chapter 11
General Type-2 Fuzzy Systems

11.1 Introduction

This chapter is about general type-2 (GT2) fuzzy systems in which all of the fuzzy sets are general type-2 fuzzy sets (GT2 FSs). Strictly speaking, only one of the fuzzy sets that are associated with a GT2 fuzzy system has to be a GT2 FS for the resulting fuzzy system to be called a GT2 fuzzy system. Unlike IT2 fuzzy systems, which for more than 17 years (as of the year 2017) have received a lot of attention, both theoretically and in real-world applications, GT2 fuzzy systems are in their infancy. And so, quite a bit of what is in this chapter is yet untried (i.e., speculative) in real-world applications, and is therefore subject to future research scrutiny.

Why are GT2 FSs and fuzzy systems needed? Here are three reasons:

1. A GT2 FS is the next logical FS model to use in a fuzzy system when satisfactory system performance cannot be achieved by using a T1 or an IT2 fuzzy system.
2. A GT2 FS is a more flexible uncertainty model than is an IT2 FS, because a GT2 FS weights uncertainty nonuniformly, whereas an IT2 FS weights it uniformly. A good example of this is the centroid, which is a type-1 interval fuzzy number for an IT2 FS but is a type-1 FS for a GT2 FS.
3. A GT2 FS can resolve statements that are semantically incompatible whereas an IT2 FS cannot. The following example is given in Greenfield and John (2009): In crisp logic, the statement $S = \{$*The perpetrator is tall.*$\}$ is equivalent to $S_{\text{crisp}} = \{$*'The perpetrator is tall' is true*$\}$. In type-1 fuzzy logic, the statement S can take the following alternative forms (only a small number of them are listed):

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6_11

- $S_{\text{type}-1}$ = {*'The perpetrator is tall' has a truth value of 0.8*}
- $S'_{\text{type}-1}$ = {*'The perpetrator is tall' has a truth value of 0.5*}

  …
- $S''_{\text{type}-1}$ = {*'The perpetrator is tall' has a truth value of 0.2*}

In IT2 fuzzy logic, the statement S can take the following alternative forms (again, only a small number of them are listed):

- S_{IT2} = {*The statement* {*'The perpetrator is tall' has a truth value of 0.8*} *has a truth value of 1.*}
- S'_{IT2} = {*The statement* {*'The perpetrator is tall' has a truth value of 0.5*} *has a truth value of 1.*}

  …
- S''_{IT2} = {*The statement* {*'The perpetrator is tall' has a truth value of 0.2*} *has a truth value of 1.*}

According to Greenfield and John (2009), the statements S_{IT2}, S'_{IT2} and S''_{IT2} are incompatible because the same truth value of unity is assigned to each of the type-1 statements. Using GT2 fuzzy logic, the following statements for S_{GT2}, S'_{GT2} and S''_{GT2} are made compatible by assigning different truth values to each of the type-1 statements.:

- S_{GT2} = {*The statement* {*'The perpetrator is tall' has a truth value of 0.8*} *has a truth value of 1*}.
- S'_{GT2} = {*The statement* {*'The perpetrator is tall' has a truth value of 0.5*} *has a truth value of 0.6*}.

  …
- S''_{GT2} = {*The statement* {*'The perpetrator is tall' has a truth value of 0.2*} *has a truth value of 0.1.*}

As for an IT2 fuzzy system, there can be two kinds of GT2 fuzzy systems, one that includes type-reduction followed by (+) defuzzification, and one that uses direct defuzzification. Both kinds of GT2 fuzzy systems (see Fig. 11.1) are described in this chapter.

Observe that the block diagrams for the GT2 fuzzy system with or without direct defuzzification look exactly like the Fig. 9.1 block diagrams for an IT2 fuzzy system. Of course, the difference between them is that in Fig. 11.1 the fuzzy sets are GT2 FSs, whereas in Fig. 9.1 they are IT2 FSs.

As in Chap. 9, it is a specific value of $\mathbf{x}$, namely $\mathbf{x}'$, that excites the GT2 fuzzy system, so that the crisp output y depends on it as $f(\mathbf{x}')$. Exactly what this nonlinear function is will be established in this chapter for the two kinds of GT2 fuzzy systems.

Because there are three ways to represent a GT2 FS (vertical slice, wavy slice, and horizontal slice), this could be an extremely tedious chapter if results were presented for all representations. Instead, our focus is on the horizontal-slice

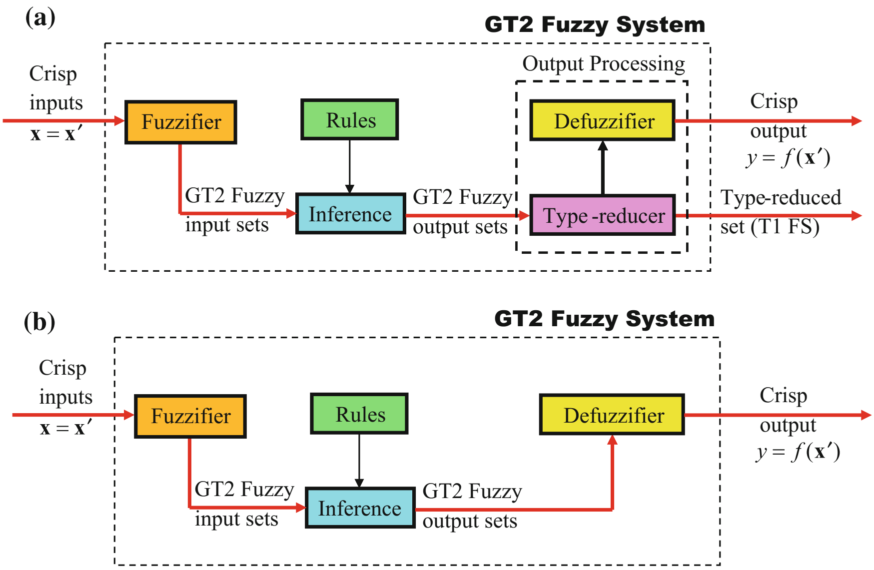


Fig. 11.1 GT2 fuzzy system with: **a** type-reduction + defuzzification, and **b** direct defuzzification

representation because it lets one use everything just learned about IT2 FSs and systems, which provides Chaps. 9 and 10 with added value.

Based on the horizontal-slice representation of a GT2 FS in Sect. 6.7.3, a horizontal-slice fuzzy system is defined, as:

Definition 11.1 A *horizontal-slice fuzzy system* is analogous to an IT2 fuzzy system where all of the IT2 FS computations that are given in Chap. 9 occur for each horizontal slice.

Definition 11.2 A *WH GT2 fuzzy system* is an aggregation of $k_{\max}$ horizontal-slice fuzzy systems, as in Fig. 11.2, where aggregation occurs by means of defuzzification.

The idea of aggregating horizontal-slice fuzzy systems was proposed originally in[1] Wagner and Hagras (2008, 2010, 2013), and was expounded upon in Mendel (2014). It is based on the horizontal-slice decomposition of a GT2 FS that is given in Theorem 6.4 and the statement in Sect. 7.12, that: α-planes of a function of GT2 FSs should equal that function applied to the α-planes of those GT2 FSs. It is referred to in this chapter as the *WH GT2 fuzzy system*, so as to distinguish it from other kinds of GT2 fuzzy systems that may be developed in the future.

The structure of this chapter parallels the structure of Chap. 9 wherever possible. In this way, a reader will be able to observe what changes, and how it changes,

[1]In the Wagner and Hagras references, the term "zSlice" is used instead of horizontal-slice. See Sect. 6.7.3 for a discussion about why this book uses "horizontal-slice" instead of zSlice.

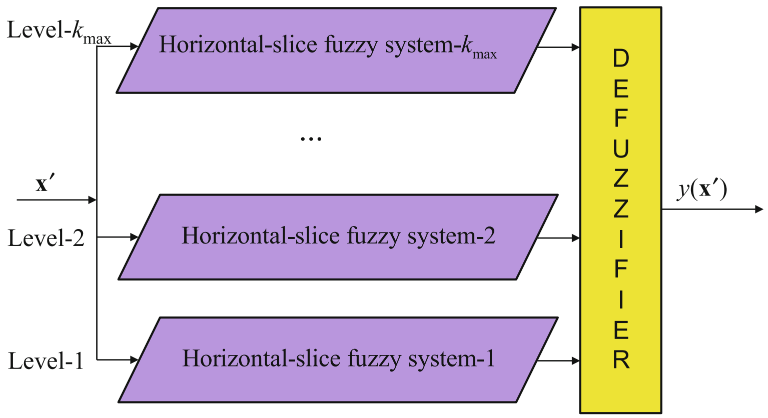


Fig. 11.2 WH GT2 fuzzy system is the aggregation of horizontal-slice IT2 fuzzy systems

as one goes from an IT2 fuzzy system to a WH GT2 fuzzy system. It also briefly covers some aspects of the designs of WH GT2 fuzzy systems, "brief" because as of 2017 not much has actually been published yet about ways to design them. It concludes with a brief survey of some recent application papers and a case study on WH GT2 FLC. A chapter like Chap. 10 is not included because, except for FLC, none of the applications and other case studies of that chapter have been extended to WH GT2 fuzzy systems. No doubt, in future years others will do this and report on their results with the same amount of detail as in Chap. 10 so that their results can be replicated.

11.2 Rules

Just as the rules of an IT2 fuzzy system can have two different canonical structures, Zadeh and TSK, rules of a GT2 fuzzy system can also have these two different structures. The distinction between IT2 and GT2 is associated with the nature of the MFs, which is not important when forming the rules. The structure of the rules remains exactly the same in the GT2 case, but now some or all of the sets involved are GT2. However, in order to distinguish between rules that use IT2 or GT2 FSs, the latter will be called *GT2 rules*.

As in an IT2 fuzzy system, a GT2 fuzzy system has p inputs $x_1 \in X_1, \ldots, x_p \in X_p$, and one output $y \in Y$, but each x_i is now described by Q_i linguistic terms that are modeled as GT2 FSs ($T_{x_i} = \{\tilde{X}_{ij}\}_{j=1}^{Q_i}$) and y is either described by Q_y linguistic terms that are modeled as GT2 FSs ($T_y = \{\tilde{Y}_j\}_{j=1}^{Q_y}$) or by a function of $x_1, \ldots,$ and x_p.

Definition 11.3 The structure of the lth generic GT2 *Zadeh rule* for a GT2 fuzzy system is $(l = 1, \ldots, M)$:

$$\tilde{R}_Z^l : \text{ IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \tilde{F}_p^l, \text{ THEN } y \text{ is } \tilde{G}^l \tag{11.1}$$

whereas the structure of the lth generic GT2 *Takagi, Sugeno, and Kang* (*TSK*) *rule* for a GT2 fuzzy system is $(l = 1, \ldots, M)$:

$$\tilde{R}_{TSK}^l : \text{ IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \tilde{F}_p^l, \text{ THEN } y \text{ is } g^l(x_1, \ldots, x_p) \tag{11.2}$$

In both (11.1) and (11.2), $\tilde{F}_1^l \in T_{x_1}$, …, and $\tilde{F}_p^l \in T_{x_p}$. In (11.1), because $\tilde{G}^l \in T_y$ is a GT2 FS, it is described by its MF $\mu_{\tilde{G}^l}(y, u)$. In (11.2), although y does not seem to be a fuzzy set it can be modeled as a *type-2 fuzzy singleton* (see Definition 6.26) $\tilde{G}^l$, so it is made to resemble a GT2 Zadeh rule, where

$$\mu_{\tilde{G}^l(y)}(u) \equiv \begin{cases} 1/1 & \text{when } y = g^l(\mathbf{x}) \\ 1/0 & \text{otherwise} \end{cases} \tag{11.3}$$

In (11.3), $\mathbf{x} = \text{col}(x_1, \ldots, x_p)$.

Comparing Definitions 11.3 and 9.1, it is a fact, as mentioned above, that the structures of generic IT2 and GT2 Zadeh and TSK rules are exactly the same. Even the symbols that are used for their different kinds of T2 FSs are the same, e.g., $\tilde{F}_j^l$. It is only the way in which these T2 FSs are modeled that distinguishes the GT2 and IT2 rules. In fact, one could just refer to both kinds of rules as “T2 rules.” The preference here is not to do this, in deference to the decomposition of the field of T2 FSs as the union of the subfields of GT2 FSs and IT2 FSs, which was explained in Sect. 6.6 (Fig. 6.18).

As in Chaps. 3 and 9, the rules in (11.1) and (11.2) are *complete IF rules* because all p inputs are present in their antecedents. These generic rules represent GT2 fuzzy relations between the input space $X_1 \times \cdots \times X_p$ and the output space, Y, of the GT2 fuzzy system, and which rule structure to use is, as is true for an IT2 fuzzy system, very much application dependent.

Definition 11.4 When a fuzzy system uses GT2 Zadeh rules and a Mamdani implication operator it will be referred to as a GT2 *Mamdani fuzzy system* (or as a *WH GT2 Mamdani fuzzy system*).

Definition 11.5 : When a fuzzy system uses GT2 TSK rules and a Mamdani implication operator it will be referred to as a GT2 *TSK fuzzy system* (or as a *WH GT2 TSK fuzzy system*).

11.3 Fuzzifier

For an IT2 fuzzy system, three kinds of fuzzifiers were established: singleton, type-1 non-singleton, and IT2 non-singleton. For a GT2 fuzzy system, in principle, four kinds of fuzzifiers are possible: singleton, type-1 non-singleton, IT2 non-singleton and GT2 non-singleton. See Definitions 9.4, 9.5, and 9.6 for the definitions of singleton, type-1 non-singleton, and IT2 non-singleton fuzzifiers.

Definition 11.6 A *GT2 non-singleton fuzzifier* maps measurement $x_i = x_i'$ into a GT2 fuzzy number.

What exactly is a GT2 fuzzy number? Hamrawi and Coupland (2009) define three kinds of GT2 fuzzy numbers:

- A *perfectly normal GT2 fuzzy number* is a GT2 FS whose FOU is a perfectly normal IT2 fuzzy number [i.e., an IT2 FS both of whose lower and upper MFs of its FOU are type-1 fuzzy numbers (see Definition 2.5)] and whose $\alpha = 1$ plane is either a type-1 fuzzy number or a normal IT2 fuzzy number (an IT2 FS only whose UMF of its FOU is a type-1 fuzzy number).
- A *normal GT2 fuzzy number* is a GT2 FS whose FOU is a normal IT2 fuzzy number and whose $\alpha = 1$ plane exists.
- A *partially normal GT2 fuzzy number* is a GT2 FS whose FOU is a normal IT2 fuzzy number.

Recall (Sect. 6.5) that there is no consensus on even what an IT2 fuzzy number is; it was defined in Definition 6.22 in what to this author is an intuitively plausible way (i.e., an IT2 fuzzy number is an IT2 FS whose lower and upper MFs of its FOU are type-1 fuzzy numbers). Because an IT2 fuzzy number must be a special case of a GT2 fuzzy number, it is only a perfectly normal GT2 fuzzy number that allows for this. Consequently, to this author, it is only a perfectly normal GT2 fuzzy number that deserves the designation of a GT2 fuzzy number.

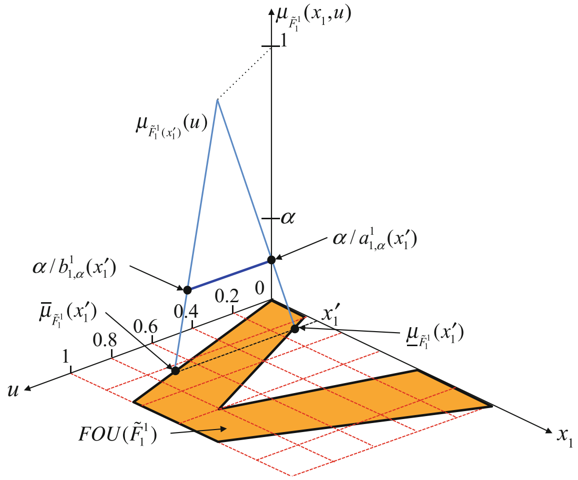
GT2 non-singleton fuzzification is not considered in the rest of this chapter. Much more research is needed to see if or when it is needed.

11.4 Fuzzy Inference Engine

The focus of this section is on results just for singleton fuzzification and convex and normal secondary MFs. Results for type-1 and IT2 non-singleton fuzzification are left for the exercises (Exercise 11.1).

Recall (Theorem 6.1) that the α-cut of the type-1 secondary MF $\tilde{A}(x)$, $\tilde{A}(x)_\alpha$, is given by $[a_\alpha(x), b_\alpha(x)]$, so that (for convex secondary MFs):

Fig. 11.3 In singleton fuzzification, the triangle secondary MF is activated when $x_1 = x'_1$



$$\tilde{A}(x) = \sup_{\alpha \in [0,1]} [\alpha/[a_\alpha(x), b_\alpha(x)]] \tag{11.4}$$

Recall, also, that in singleton fuzzification, the meet between the input and its antecedent consequent GT2 FS is a T1 FS (see the Sifting Theorem 7.7). So, for each rule, when $x_i = x'_i$ only the vertical slice of the rule-antecedent GT2 FS $\tilde{F}_i^l$, $\tilde{F}_i^l(x'_i)$, is activated, and it has the following α-cut decomposition[2] (Fig. 11.3) $(i = 1, \ldots, p \text{ and } l = 1, 2, \ldots, M)$:

$$\tilde{F}_i^l(x'_i) \Leftrightarrow \mu_{\tilde{F}_i^l(x'_i)}(u) = \sup_{\alpha \in [0,1]} \alpha/[a^l_{i,\alpha}(x'_i), b^l_{i,\alpha}(x'_i)] \tag{11.5}$$

Definition 11.7 For a GT2 Mamdani or TSK fuzzy system, the *level α firing set*, $F^l_\alpha(\mathbf{x}')$, is $(l = 1, \ldots, M$ and $\alpha \in [0, 1])$:

$$F^l_\alpha(\mathbf{x}') \equiv \alpha/[\underline{f}^l_\alpha(\mathbf{x}'), \bar{f}^l_\alpha(\mathbf{x}')] \tag{11.6}$$

In (11.6) $[\underline{f}^l_\alpha(\mathbf{x}'), \bar{f}^l_\alpha(\mathbf{x}')]$ is the *level α firing interval* at $\mathbf{x} = \mathbf{x}'$, and

$$\underline{f}^l_\alpha(\mathbf{x}') \equiv T^p_{i=1} a^l_{i,\alpha}(x'_i) \tag{11.7}$$

$$\bar{f}^l_\alpha(\mathbf{x}') \equiv T^p_{i=1} b^l_{i,\alpha}(x'_i) \tag{11.8}$$

[2]In a GT2 fuzzy system, in (11.5) and other places, one needs to keep track of which antecedent is referred to (the first subscript i), which rule is referred to (the superscript l), and which α-cut is referred to (the second subscript α); this unavoidably leads to very heavy subscript and superscript notations.

Definition 11.8 $\tilde{B}^l_\alpha$ is called a *horizontal-slice* (*level* α) *fired-rule output set*.

Theorem 11.1 *For singleton fuzzification, the MF* $\mu_{\tilde{B}^l_\alpha}$ *for a horizontal-slice fired-rule output set* $\tilde{B}^l_\alpha$ *is* ($y \in Y$, $l = 1, \ldots, M$ and $\alpha \in [0, 1]$):

$$\begin{cases} \text{WH Mamdani fuzzy system:} \quad \mu_{\tilde{B}^l_\alpha(y|\mathbf{x}')} = \alpha/[\underline{\mu}_{\tilde{B}^l_\alpha}(y|\mathbf{x}'), \bar{\mu}_{\tilde{B}^l_\alpha}(y|\mathbf{x}')] = \alpha/\mathrm{FOU}(\tilde{B}^l_\alpha) \\ \text{WH TSK fuzzy system:} \quad \mu_{\tilde{B}^l_\alpha(\mathbf{x}')} = \alpha/[\underline{f}^l_\alpha(\mathbf{x}'), \bar{f}^l_\alpha(\mathbf{x}')] \quad \text{when} \quad y = g^l(\mathbf{x}') \end{cases} \tag{11.9}$$

For the WH Mamdani fuzzy system:

$$\tilde{G}^l_\alpha = \int_{y \in Y} G^l_\alpha(y)/y = \int_{y \in Y} [c^l_\alpha(y), d^l_\alpha(y)]/y \tag{11.10}$$

$$\underline{\mu}_{\tilde{B}^l_\alpha}(y|\mathbf{x}') = \underline{f}^l_\alpha(\mathbf{x}') \bigstar c^l_\alpha(y) \tag{11.11}$$

$$\bar{\mu}_{\tilde{B}^l_\alpha}(y|\mathbf{x}') = \bar{f}^l_\alpha(\mathbf{x}') \bigstar d^l_\alpha(y) \tag{11.12}$$

Proof This theorem is the horizontal-slice version of Corollaries 9.1 and 9.2.

Observe, from the top line of (11.9), that $\mathrm{FOU}(\tilde{B}^l_\alpha)$ is a plane on $Y \times U$ in a WH GT2 Mamdani fuzzy system.

Definition 11.9 A *horizontal-slice first-order rule partition* of $X_1 \times X_2 \times \cdots \times X_p$ is a collection of nonoverlapping rectangles at level-α ($\alpha \in [0, 1]$) in each of which a fixed number of rules is fired, where that number is found by examining the horizontal slices (at level-α) of the antecedent GT2 FSs simultaneously (e.g., see Fig. 6.24 for some horizontal slices).

Definition 11.10 A *horizontal-slice second-order rule partition* of $X_1 \times X_2 \times \cdots \times X_p$ occurs when the LMF or the UMF of the horizontal-slice (α-plane at level-α) that is associated with either x_1, or x_2, or …, or x_p, changes its mathematical formula within a horizontal-slice first-order rule partition.

It is conjectured that for closed GT2 FSs (Definition 6.25) each horizontal-slice first-order rule partition is the same for all values of $\alpha \in [0, 1]$, and so one only has to focus on the $\alpha = 0$ first-order rule partitions (i.e., the first-order rule partitions of the FOUs of the GT2 FSs) of $X_1 \times X_2 \times \cdots \times X_p$. However, horizontal-slice second-order rule partitions may be different for different values of α. It is these differences that provide a GT2 fuzzy system with additional degrees of freedom over an IT2 fuzzy system. What these partitions look like, as well as how many there are of them, remain to be explored.

Example 11.1 This is a two-rule example (see Exercise 11.2) that is analogous to Example 9.3. The rules are GT2 Zadeh rules:

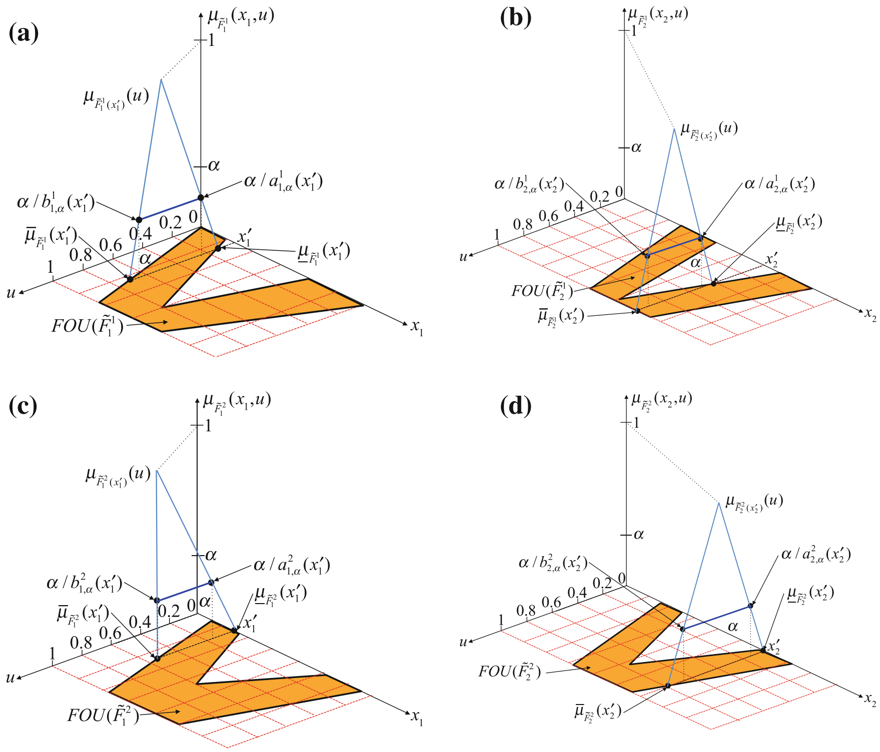


Fig. 11.4 Antecedent MFs for Example 11.1. **a** and **b** are for $\tilde{R}_Z^1$, and **c** and **d** are for $\tilde{R}_Z^2$

$$\begin{aligned} \tilde{R}_Z^1 &: \text{ IF } x_1 \text{ is } \tilde{F}_1^1 \text{ and } x_2 \text{ is } \tilde{F}_2^1 \text{ THEN } y \text{ is } \tilde{G}^1 \\ \tilde{R}_Z^2 &: \text{ IF } x_1 \text{ is } \tilde{F}_1^2 \text{ and } x_2 \text{ is } \tilde{F}_2^2 \text{ THEN } y \text{ is } \tilde{G}^2 \end{aligned} \tag{11.13}$$

All of the fuzzy sets are GT2 FSs and have triangle secondary MFs; their FOUs are depicted in Fig. 11.4. Each of the four parts to Fig. 11.4 also depicts the secondary MF that is sifted out when $x_1 = x_1'$ and $x_2 = x_2'$, and they show a specific α-cut raised to level-α (α is not a variable in the rest of this example), namely:

$$\begin{aligned} &\text{(a) } \alpha/[a_{1,\alpha}^1(x_1'), b_{1,\alpha}^1(x_1')], \text{(b) } \alpha/[a_{2,\alpha}^1(x_2'), b_{2,\alpha}^1(x_2')], \\ &\text{(c) } \alpha/[a_{1,\alpha}^2(x_1'), b_{1,\alpha}^2(x_1')] \text{ and (d) } \alpha/[a_{2,\alpha}^2(x_2'), b_{2,\alpha}^2(x_2')]. \end{aligned}$$

Formulas for the level-α firing sets for the two rules are (11.6)–(11.8):

$$\begin{cases} \tilde{F}_\alpha^1(x_1', x_2') = \alpha/[\underline{f}_\alpha^1(x_1', x_2'), \bar{f}_\alpha^1(x_1', x_2')] = \alpha/[a_{1,\alpha}^1(x_1') \bigstar a_{2,\alpha}^1(x_2'), b_{1,\alpha}^1(x_1') \bigstar b_{2,\alpha}^1(x_2')] \\ \tilde{F}_\alpha^2(x_1', x_2') = \alpha/[\underline{f}_\alpha^2(x_1', x_2'), \bar{f}_\alpha^2(x_1', x_2')] = \alpha/[a_{1,\alpha}^2(x_1') \bigstar a_{2,\alpha}^2(x_2'), b_{1,\alpha}^2(x_1') \bigstar b_{2,\alpha}^2(x_2')] \end{cases} \tag{11.14}$$

The end points of the α-cuts needed to compute $\tilde{F}^1_\alpha(x'_1, x'_2)$ and $\tilde{F}^2_\alpha(x'_1, x'_2)$ can be read off of the four parts of Fig. 11.4 (so they are approximate), and using the dashed vertical lines from the α-cuts raised to level-α down to the FOU, are:

$$\begin{cases} \tilde{R}^1_Z : a^1_{1,\alpha}(x'_1) \approx 0.23,\ b^1_{1,\alpha}(x'_1) \approx 0.64,\ a^1_{2,\alpha}(x'_2) \approx 0.5 \quad \text{and} \quad b^1_{2,\alpha}(x'_2) \approx 0.92 \\ \tilde{R}^2_Z : a^2_{1,\alpha}(x'_1) \approx 0.15,\ b^2_{1,\alpha}(x'_1) \approx 0.54,\ a^2_{2,\alpha}(x'_2) \approx 0.13 \quad \text{and} \quad b^2_{2,\alpha}(x'_2) \approx 0.63 \end{cases} \tag{11.15}$$

Using the minimum *t*-norm in (11.14), it follows therefore that:

$$\begin{cases} \tilde{R}^1_Z : \underline{f}^1_\alpha(x'_1, x'_2) = \min(0.23, 0.5) = 0.23 \quad \text{and} \quad \bar{f}^1_\alpha(x'_1, x'_2) = \min(0.64, 0.92) = 0.64 \\ \tilde{R}^2_Z : \underline{f}^2_\alpha(x'_1, x'_2) = \min(0.15, 0.13) = 0.13 \quad \text{and} \quad \bar{f}^2_\alpha(x'_1, x'_2) = \min(0.54, 0.63) = 0.54 \end{cases} \tag{11.16}$$

so that:

$$\begin{cases} \tilde{R}^1_Z : \tilde{F}^1_\alpha(x'_1, x'_2) = \alpha/[0.23, 0.64] \\ \tilde{R}^2_Z : \tilde{F}^2_\alpha(x'_1, x'_2) = \alpha/[0.13, 0.54] \end{cases} \tag{11.17}$$

For the consequents of $\tilde{R}^1_Z$ and $\tilde{R}^2_Z$, Fig. 11.5 depicts FOUs, triangle secondary MFs, and α-planes raised to level-α, $\alpha/\tilde{G}^1_\alpha$ and $\alpha/\tilde{G}^2_\alpha$, respectively (see Exercise 11.3).

Figure 11.6 depicts FOU$(\tilde{B}^1_\alpha)$ and FOU$(\tilde{B}^2_\alpha)$. More specifically, Fig. 11.6a was constructed by first plotting $\tilde{G}^1_\alpha$ (the orange FOU), where its coordinates were

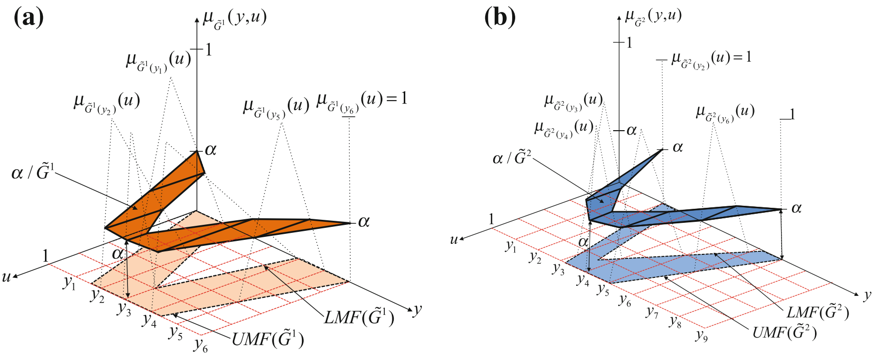


Fig. 11.5 Consequent MFs for Example 11.1. **a** is for $\tilde{G}^1$ in $\tilde{R}^1_Z$, and **b** is for $\tilde{G}^2$ in $\tilde{R}^2_Z$

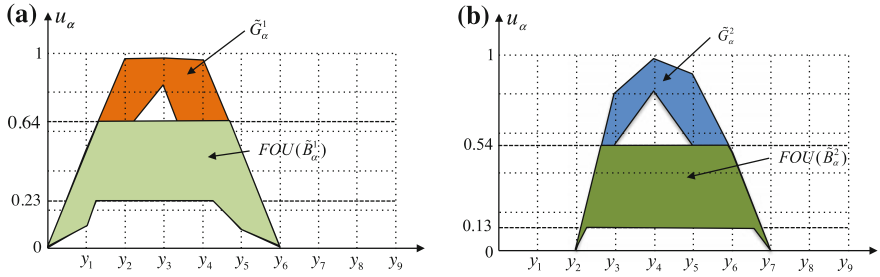


Fig. 11.6 **a** $\mathrm{FOU}(\tilde{B}^1_\alpha)$ and **b** $\mathrm{FOU}(\tilde{B}^2_\alpha)$. u_α denotes the secondary variable at level α

picked off of Fig. 11.5a. The two horizontal lines are from (11.17) for $\tilde{R}^1_Z$, namely $\underline{f}^1_\alpha(x'_1, x'_2) = 0.23$ and $\bar{f}^1_\alpha(x'_1, x'_2) = 0.64$. Then the LMF and UMF of the light green FOU were found from (11.11) and (11.12), as $(y \in Y)$:

$$\begin{cases} \underline{\mu}_{\tilde{B}^1_\alpha}(y|x'_1, x'_2) = \underline{f}^1_\alpha(x'_1, x'_2) \bigstar c^1_\alpha(y) = \min(0.23, c^1_\alpha(y)) \\ \bar{\mu}_{\tilde{B}^1_\alpha}(y|x'_1, x'_2) = \bar{f}^1_\alpha(x'_1, x'_2) \bigstar d^1_\alpha(y) = \min(0.64, d^1_\alpha(y)) \end{cases} \tag{11.18}$$

In (11.18), $c^1_\alpha(y)(y \in Y)$ defines the LMF of $\tilde{G}^1_\alpha$ and $d^1_\alpha(y)(y \in Y)$ defines the UMF of $\tilde{G}^1_\alpha$. Figure 11.6b was constructed in a similar manner, using Fig. 11.5b and (11.17) for $\tilde{R}^2_Z$.

In a WH Mamdani fuzzy system, all of these computations would be performed at $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_{k_{\max}}$.

11.5 Combining Fired Rule Output Sets on the Way to Defuzzification

Observe in Fig. 11.1 that the output(s) from the inference engine block go either to the type-reduction block or directly to the defuzzifier block. If more than one rule fires, one first needs to decide what to do with them (as was the situation for an IT2 fuzzy system) either on the way to type-reduction or to direct defuzzification. As pointed out in Sect. 3.5, there is no unique way to combine fired-rule output sets. In the rest of this section, two possibilities are described.

11.5.1 Combining Using Set Theoretic Operations in a WH GT2 Mamdani Fuzzy System

Just as IT2 fired-rule output sets for an IT2 Mamdani fuzzy system can be combined by using the union operation, as explained in Sect. 9.5.1, GT2 fired-rule output sets for a GT2 Mamdani fuzzy system can also be combined by using the union operation. In a WH GT2 fuzzy system this is done for each horizontal slice.

Theorem 11.2 *Suppose that M_F of the M rules in a WH GT2 Mamdani fuzzy system fire, where $M_F \le M$, and a combined output GT2 FS $\tilde{B}$ is obtained by combining the fired-rule output sets using the union operation. Then for singleton fuzzification, the* horizontal-slice aggregated output *is* ($y \in Y$ and $\alpha \in [0,1]$):

$$\mu_{\tilde{B}_\alpha(y|\mathbf{x}')} = \alpha/[\underline{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}'), \bar{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}')] = \alpha/\mathrm{FOU}(\tilde{B}_\alpha(y|\mathbf{x}')) \tag{11.19}$$

where

$$\underline{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}') = \underline{\mu}_{\tilde{B}^1_\alpha}(y|\mathbf{x}') \vee \underline{\mu}_{\tilde{B}^2_\alpha}(y|\mathbf{x}') \vee \cdots \vee \underline{\mu}_{\tilde{B}^{M_F}_\alpha}(y|\mathbf{x}') \tag{11.20}$$

$$\bar{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}') = \bar{\mu}_{\tilde{B}^1_\alpha}(y|\mathbf{x}') \vee \bar{\mu}_{\tilde{B}^2_\alpha}(y|\mathbf{x}') \vee \cdots \vee \bar{\mu}_{\tilde{B}^{M_F}_\alpha}(y|\mathbf{x}') \tag{11.21}$$

In (11.20) and (11.21), $(l = 1, \ldots, M_F)$ $\underline{\mu}_{\tilde{B}^l_\alpha}(y|\mathbf{x}')$ *is computed using (11.11), and* $\bar{\mu}_{\tilde{B}^l_\alpha}(y|\mathbf{x}')$ *is computed using (11.12).* The aggregated output of a WH GT2 Mamdani fuzzy system, $\tilde{B}_{\mathrm{WH}}$, *is* $(y \in Y)$:

$$\tilde{B}_{\mathrm{WH}} = \bigcup_{\alpha\in[0,1]} \alpha/\mathrm{FOU}(\tilde{B}_\alpha(y|\mathbf{x}')) = \sup_{\alpha\in[0,1]} \alpha/\mathrm{FOU}(\tilde{B}_\alpha(y|\mathbf{x}')) \tag{11.22}$$

Proof This theorem is the horizontal-slice version of Theorem 9.2.

Definition 11.11 For a GT2 Mamdani fuzzy system $\mathrm{FOU}(\tilde{B}_\alpha(y|\mathbf{x}'))$—$\mathrm{FOU}(\tilde{B}_\alpha)$ for short—is called the *FOU of the horizontal-slice aggregated output*.

Example 11.2 Figure 11.7 depicts $\mathrm{FOU}(\tilde{B}_\alpha)$ for the combined output set, $\tilde{B}$, for the Example 11.1 two-rule GT2 fuzzy system, where the fired-rule horizontal slices $\mathrm{FOU}(\tilde{B}^1_\alpha)$ and $\mathrm{FOU}(\tilde{B}^2_\alpha)$ are depicted in Fig. 11.6a, b, respectively, and are combined as in (11.19)–(11.21) using the maximum for the disjunction. The easiest way to verify $\mathrm{FOU}(\tilde{B}_\alpha)$ is to put $\mathrm{FOU}(\tilde{B}^1_\alpha)$ and $\mathrm{FOU}(\tilde{B}^2_\alpha)$ on the same plot and then draw $(y \in Y)$ $\underline{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}')$ as $\max[\underline{\mu}_{\tilde{B}^1_\alpha}(y|\mathbf{x}'), \underline{\mu}_{\tilde{B}^2_\alpha}(y|\mathbf{x}')]$, and $\bar{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}')$ as $\max[\bar{\mu}_{\tilde{B}^1_\alpha}(y|\mathbf{x}'), \bar{\mu}_{\tilde{B}^2_\alpha}(y|\mathbf{x}')]$.

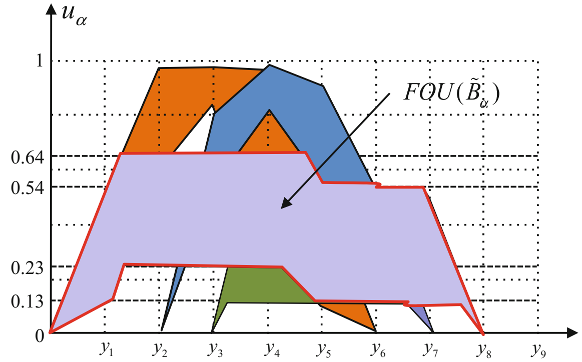


Fig. 11.7 Pictorial description of the combined two-rule fired-rule output sets depicted in Fig. 11.6a, b. u_α denotes the secondary variable at levelα

11.5.2 Combining During Defuzzification in a WH GT2 Mamdani Fuzzy System

In principle one can compute the aggregated output of a WH GT2 Mamdani fuzzy system, $\tilde{B}_{WH}$, as in (11.22), however, when a WH GT2 Mamdani fuzzy system is used in a real-world application, the issue of computational complexity becomes as important as it is when an IT2 Mamdani fuzzy system is used, especially if the application is one that requires real-time processing. For a WH GT2 Mamdani fuzzy system, taking the fuzzy union of horizontal-slice aggregated outputs requires additional computation time and storage, which may be very undesirable, and so an alternative to performing any kind of combining prior to defuzzification for such rules is to perform no combining. What this means is explained next in Sect. 11.6.

11.6 Type-Reduction

Recall, from Sect. 9.6, that three kinds of type-reduction were described for an IT2 Mamdani fuzzy system (centroid, height and center-of-sets type-reduction) and type-reduction was also described for four kinds of IT2 TSK fuzzy systems. You will note that Sect. 9.6 is called “Type-Reduction + Defuzzification,” but the present section is only titled “Type-Reduction.” The reason for this is that, whereas defuzzification after type-reduction is trivial for an IT2 fuzzy system (just average the end-points of the type-reduced set), the same is not true for a WH GT2 fuzzy system. As will be seen below, type-reduction[3] for a WH GT2 fuzzy system is

[3]Almaraashi et al. (2016) perform type-reduction followed by defuzzification by first computing the COG of vertical slices (some interpolation of the COGs is also used) and then defuzzifying the resulting T1 FS. Although this kind of type-reduction is ad hoc, it does adhere to the basic design requirement that when all MF uncertainties disappear the output of a T2 fuzzy system must become the same as the output of a T1 fuzzy system. If one is only interested in the defuzzified

performed for each horizontal slice after which the type-reduced results are aggregated across all of the horizontal slices by means of defuzzification. *It is the defuzzifier that couples all of the horizontal-slice fuzzy systems* (see Fig. 11.2).

Definition 11.12 *Horizontal-slice type-reduction* is type-reduction applied to horizontal-slice quantities, the result being a *horizontal-slice type-reduced set*.

Different kinds of type-reduction use different horizontal-slice quantities.

There can be different ways to perform defuzzification in a WH GT2 fuzzy system, and so instead of ending each of the subsections of this section with a defuzzification formula, as was done in the subsections of Sect. 9.6, defuzzification is covered separately in Sect. 11.7.

11.6.1 Centroid Type-Reduction for a WH GT2 Mamdani Fuzzy System

Centroid type-reduction for a WH GT2 Mamdani fuzzy system is performed separately for each horizontal slice. Theorem 8.4 is the basis for doing this. Stated in terms of $\tilde{B}_{WH}$ in (11.22), (8.48) is:

$$C_{\tilde{B}_{WH}}(y) = \bigcup_{\alpha\in[0,1]} C_{R_{\tilde{B}_\alpha}}(y) = \bigcup_{\alpha\in[0,1]} \alpha/[c_l(R_{\tilde{B}_\alpha}), c_r(R_{\tilde{B}_\alpha})] \equiv \bigcup_{\alpha\in[0,1]} \alpha/[c_l(\alpha|\mathbf{x}'), c_r(\alpha|\mathbf{x}')] \tag{11.23}$$

In (11.23), $C_{R_{\tilde{B}_\alpha}}(y)$ is the centroid of $\alpha/\mathrm{FOU}(\tilde{B}_\alpha)$ (i.e., the *horizontal-slice centroid*), and $c_l(\alpha|\mathbf{x}')$ and $c_r(\alpha|\mathbf{x}')$ are computed as explained in Theorem 8.1, in which:

$$c_l(\alpha|\mathbf{x}') = \frac{\sum_{i=1}^{L} y_i \bar{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}') + \sum_{i=L+1}^{N} y_i \underline{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}')}{\sum_{i=1}^{L} \bar{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}') + \sum_{i=L+1}^{N} \underline{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}')} \tag{11.24}$$

$$c_r(\alpha|\mathbf{x}') = \frac{\sum_{i=1}^{R} y_i \underline{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}') + \sum_{i=R+1}^{N} y_i \bar{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}')}{\sum_{i=1}^{R} \underline{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}') + \sum_{i=R+1}^{N} \bar{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}')} \tag{11.25}$$

In (11.24) and (11.25), $\underline{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}')$ are sampled values of $\underline{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}')$, that are given in (11.20) and (11.21), respectively. EIASC or EKM

(Footnote 3 continued)

output of a GT2 fuzzy system, and not a measure of the uncertainties that have flowed through that system, then their *vertical slice centroid type-reducer* (VSCTR) is arguably a very viable alternative to the WH approach that is taken in this chapter.

algorithms can be used to compute $c_l(\alpha|\mathbf{x}')$ and $c_r(\alpha|\mathbf{x}')$. Note that, because the domain of $\tilde{B}_\alpha$ is Y, $c_l(\alpha|\mathbf{x}')$ and $c_r(\alpha|\mathbf{x}')$ are located on the y-axis.

11.6.2 Center-of Sets Type-Reduction for a WH GT2 Mamdani Fuzzy System

Center-of sets (COS) type-reduction[4] for a WH GT2 fuzzy system is performed separately for each horizontal slice. The following statement that is given in Sect. 7.12 is the basis for this: α-planes of a function of GT2 FSs should equal that function applied to the α-planes of those GT2 FSs.

To begin, one must compute the centroids of the M GT2 Zadeh-rule consequent GT2 FSs, i.e., $(l = 1, \ldots, M)$:

$$C_{\tilde{G}^l} = \sup_{\forall\alpha\in[0,1]} C_{\tilde{G}^l_\alpha} \tag{11.26}$$

$$C_{\tilde{G}^l_\alpha} = \alpha\big/\left[c_l(\tilde{G}^l_\alpha), c_r(\tilde{G}^l_\alpha)\right] \tag{11.27}$$

where $C_{\tilde{G}^l_\alpha}$ is the centroid of α-plane $\tilde{G}^l_\alpha$ raised to level-α. After the design of the WH GT2 fuzzy system has been completed, these centroids can be computed one last time and then stored because they do not depend upon $\mathbf{x}'$.

The M $C_{\tilde{G}^l_\alpha}$ are used, along with the level-α firing sets, $F^l_\alpha(\mathbf{x}')$, to compute

$$Y_{\text{COS},\alpha}(\mathbf{x}') = \alpha/[y^{\text{COS}}_{l,\alpha}(\mathbf{x}'), y^{\text{COS}}_{r,\alpha}(\mathbf{x}')] \tag{11.28}$$

In (11.28), $y^{\text{COS}}_{l,\alpha}(\mathbf{x}')$ and $y^{\text{COS}}_{r,\alpha}(\mathbf{x}')$ are computed as explained in Theorem 8.3, as [see, also (9.111) and (9.112)]:

$$y^{\text{COS}}_{l,\alpha}(\mathbf{x}') = \frac{\sum_{i=1}^{L} c_l(\tilde{G}^l_\alpha)\bar{f}^i_\alpha(\mathbf{x}') + \sum_{i=L+1}^{M} c_l(\tilde{G}^l_\alpha)\underline{f}^i_\alpha(\mathbf{x}')}{\sum_{i=1}^{L}\bar{f}^i_\alpha(\mathbf{x}') + \sum_{i=L+1}^{M}\underline{f}^i_\alpha(\mathbf{x}')} \tag{11.29}$$

$$y^{\text{COS}}_{r,\alpha}(\mathbf{x}') = \frac{\sum_{i=1}^{R} c_r(\tilde{G}^l_\alpha)\underline{f}^i_\alpha(\mathbf{x}') + \sum_{i=R+1}^{M} c_r(\tilde{G}^l_\alpha)\bar{f}^i_\alpha(\mathbf{x}')}{\sum_{i=1}^{R}\underline{f}^i_\alpha(\mathbf{x}') + \sum_{i=R+1}^{M}\bar{f}^i_\alpha(\mathbf{x}')} \tag{11.30}$$

In (11.29) and (11.30), $\underline{f}^i_\alpha(\mathbf{x}')$ and $\bar{f}^i_\alpha(\mathbf{x}')$ are the end-points of the level-α firing interval for the ith rule, computed using (11.7) and (11.8), respectively. EIASC or

[4]Height type-reduction for a WH GT2 fuzzy system is left as an exercise (Exercise 11.4).

EKM algorithms can be used to compute $y_{l,\alpha}^{COS}(\mathbf{x}')$ and $y_{r,\alpha}^{COS}(\mathbf{x}')$, both of which are located on the y-axis.

It follows, then that

$$Y_{\mathrm{WH-COS}}(\mathbf{x}') = \bigcup_{\alpha\in[0,1]} Y_{\mathrm{COS},\alpha}(\mathbf{x}') = \bigcup_{\alpha\in[0,1]} \alpha/[y_{l,\alpha}^{\mathrm{COS}}(\mathbf{x}'), y_{r,\alpha}^{\mathrm{COS}}(\mathbf{x}')] \tag{11.31}$$

11.6.3 Type-Reduction for a[5] WH GT2 TSK Fuzzy System

Just as there can be four kinds of IT2 TSK fuzzy systems (see Sect. 9.6.4), there can be at least[6] four kinds of WH GT2 TSK fuzzy systems, namely:

1. *Unnormalized A2-C0 WH GT2 TSK fuzzy system*, in which the antecedents are GT2 FSs and the consequent is $g^l = c_0^l + c_1^l x_1 + c_2^l x_2 + \cdots + c_p^l x_p$, in which the c_i^l coefficient are numbers. No iterative computations are needed to compute the output of this fuzzy system.
2. *Normalized A2-C0 WH GT2 TSK fuzzy system*, in which the antecedents are GT2 FSs and the consequent is $g^l = c_0^l + c_1^l x_1 + c_2^l x_2 + \cdots + c_p^l x_p$, in which the c_i^l coefficient are numbers. Iterative computations are needed to compute the output of this fuzzy system.
3. *Unnormalized A2-C1 WH GT2 TSK fuzzy system* in which the antecedents are GT2 FSs and the consequent is $G^l = C_0^l + C_1^l x_1 + C_2^l x_2 + \cdots + C_p^l x_p$, in which the C_i^l coefficients are *type-1 interval fuzzy numbers*. No iterative computations are needed to compute the output of this fuzzy system.
4. *Normalized A2-C1 WH GT2 TSK fuzzy system*, in which the antecedents are GT2 FSs and the consequent is $G^l = C_0^l + C_1^l x_1 + C_2^l x_2 + \cdots + C_p^l x_p$, in which the C_i^l coefficients are *type-1 interval fuzzy numbers*. Iterative computations are needed to compute the output of this fuzzy system.

This section explains type-reduction only for two of these GT2 TSK fuzzy systems, namely the unnormalized and normalized A2-C0 GT2 TSK fuzzy systems. Type-reductions for the unnormalized and normalized A2-C1 GT2 TSK fuzzy systems are left as exercises for the reader (Exercise 11.5a, b).

[5]Strictly speaking, Wagner and Hagras (2008, 2010, 2013) are only for Mamdani GT2 fuzzy systems; however, their approach is also applicable to GT2 TSK fuzzy systems, which is why the title of this section includes "WH."

[6]"At least" is used here because it is conceivable that even some sort of GT2 FS functions could be used for the GT2 TSK rule consequents. How to do this is an open research issue.

11.6.3.1 Unnormalized A2-C0 WH GT2 TSK Fuzzy System

In this simplest of all GT2 TSK fuzzy systems, one begins by computing $Y^U_{\mathrm{TSK},\alpha}(\mathbf{x}')$, where (see Sect. 9.6.4.1)

$$Y^U_{\mathrm{TSK},\alpha}(\mathbf{x}') = \alpha/[y^U_{\mathrm{TSK},l,\alpha}(\mathbf{x}'), y^U_{\mathrm{TSK},r,\alpha}(\mathbf{x}')] \tag{11.32}$$

in which

$$y^U_{\mathrm{TSK},l,\alpha}(\mathbf{x}') = \sum_{i=1}^{M} g^i(\mathbf{x}')\underline{f}^i_\alpha(\mathbf{x}') \tag{11.33}$$

$$y^U_{\mathrm{TSK},r,\alpha}(\mathbf{x}') = \sum_{i=1}^{M} g^i(\mathbf{x}')\bar{f}^i_\alpha(\mathbf{x}') \tag{11.34}$$

$$g^i(\mathbf{x}') = c^i_0 + c^i_1 x'_1 + c^i_2 x'_2 + \cdots + c^i_p x'_p \tag{11.35}$$

and $\underline{f}^i_\alpha(\mathbf{x}')$ and $\bar{f}^i_\alpha(\mathbf{x}')$ are the end-points of the level-α firing interval for the ith rule, computed using (11.7) and (11.8), respectively. Observe that no iterative computations are needed to compute (11.33) and (11.34).

It follows then, that

$$Y^U_{\mathrm{WH-TSK}}(\mathbf{x}') = \bigcup_{\alpha\in[0,1]} Y^U_{TSK,\alpha}(\mathbf{x}') = \bigcup_{\alpha\in[0,1]} \alpha/[y^U_{\mathrm{TSK},l,\alpha}(\mathbf{x}'), y^U_{\mathrm{TSK},r,\alpha}(\mathbf{x}')] \tag{11.36}$$

11.6.3.2 Normalized A2-C0 WH GT2 TSK Fuzzy System

For this GT2 TSK fuzzy system, one begins by computing $Y^N_{TSK,\alpha}(\mathbf{x}')$, where (see Sect. 9.6.4.2)

$$\begin{aligned} Y^N_{\mathrm{TSK},\alpha}(\mathbf{x}') &= \alpha\Big/\left[y^N_{\mathrm{TSK},l,\alpha}(\mathbf{x}'), y^N_{\mathrm{TSK},r,\alpha}(\mathbf{x}')\right] \\ &= \alpha\Bigg/\int_{f^1_\alpha(\mathbf{x}')\in\left[\underline{f}^1_\alpha(\mathbf{x}'),\bar{f}^1_\alpha(\mathbf{x}')\right]} \cdots \int_{f^M_\alpha(\mathbf{x}')\in\left[\underline{f}^M_\alpha(\mathbf{x}'),\bar{f}^M_\alpha(\mathbf{x}')\right]} \frac{\sum_{i=1}^M f^i_\alpha(\mathbf{x}')g^i(\mathbf{x}')}{\sum_{i=1}^M f^i_\alpha(\mathbf{x}')} \end{aligned} \tag{11.37}$$

where

$$y^N_{\mathrm{TSK},l,\alpha}(\mathbf{x}') = \frac{\sum_{i=1}^L g^i(\mathbf{x}')\bar{f}^i_\alpha(\mathbf{x}') + \sum_{i=L+1}^M g^i(\mathbf{x}')\underline{f}^i_\alpha(\mathbf{x}')}{\sum_{i=1}^L \bar{f}^i_\alpha(\mathbf{x}') + \sum_{i=L+1}^M \underline{f}^i_\alpha(\mathbf{x}')} \tag{11.38}$$

$$y_{\mathrm{TSK},r,\alpha}^{N}(\mathbf{x}') = \frac{\sum_{i=1}^{R} g^i(\mathbf{x}') \underline{f}_{\alpha}^{i}(\mathbf{x}') + \sum_{i=R+1}^{M} g^i(\mathbf{x}') \bar{f}_{\alpha}^{i}(\mathbf{x}')}{\sum_{i=1}^{R} \underline{f}_{\alpha}^{i}(\mathbf{x}') + \sum_{i=R+1}^{M} \bar{f}_{\alpha}^{i}(\mathbf{x}')} \tag{11.39}$$

In (11.38) and (11.39), $\underline{f}_{\alpha}^{i}(\mathbf{x}')$ and $\bar{f}_{\alpha}^{i}(\mathbf{x}')$ are the end-points of the level-α firing interval for the ith rule, computed using (11.7) and (11.8), respectively. EKM or EIASC algorithms can be used to compute $y_{\mathrm{TSK},l,\alpha}^{N}(\mathbf{x}')$ and $y_{\mathrm{TSK},r,\alpha}^{N}(\mathbf{x}')$, both of which are located on the y-axis.

It follows then, that $Y_{\mathrm{WH-TSK}}^{N}(\mathbf{x}')$ is again given by (11.36), in which unnormalized quantities are replaced by their normalized counterparts.

11.7 Defuzzification

Recall, from Sect. 11.6, that it is the defuzzifier that couples all of the horizontal-slice fuzzy systems (Fig. 11.2). In this section three ways are described to perform defuzzification on any of the just-computed horizontal-slice type-reduced fuzzy sets. They are explained only for a WH GT2 Mamdani fuzzy system that uses COS type-reduction (Sect. 11.6.2); however, exactly the same explanations apply to all other WH GT2 fuzzy systems.

All three defuzzifiers begin with $Y_{\mathrm{WH-COS}}(\mathbf{x}')$ in (11.31). Interestingly, the $\alpha = 0$ slice is excluded because, as will be seen, it does not contribute anything to the defuzzified result.[7]

11.7.1 Approximation and Defuzzification

Observe, in Fig. 11.8a, that when $Y_{\mathrm{WH-COS}}(\mathbf{x}')$ is viewed as a function of y, and the two end-points of each $\alpha/Y_{\mathrm{COS},\alpha}(\mathbf{x}')$ are projected down onto the y-axis, the result is a nonuniformly sampled set of points,[8] $y_1, y_2, \ldots, y_{2k}$. In order to achieve uniform sampling along the y-axis, the $2k$ end point values of $\alpha/Y_{\mathrm{COS},\alpha}(\mathbf{x}')(\alpha = \alpha_1, \ldots, \alpha_k)$ can be approximated using a spline approximation, the result being $\hat{Y}_{\mathrm{WH-COS}}(y)$ (see Fig. 11.8b), which is then sampled uniformly (i.e., $y'_{j+1} - y'_j = \Delta$ for $\forall j$), after which the defuzzified value of $\hat{Y}_{\mathrm{WH-COS}}(y)$, $y_{\mathrm{WH-1}}(\mathbf{x}')$, is computed as its COG, i.e.,:

[7]Wagner and Hagras (2010) was the first to make this observation.

[8]Of course, when α is sampled very finely then $y_1, y_2, \ldots, y_{2k}$ will approach a uniformly sampled set of samples; however, in a WH GT2 fuzzy system one does not want to use too many horizontal slices, or else computational complexity greatly increases as does computation time.

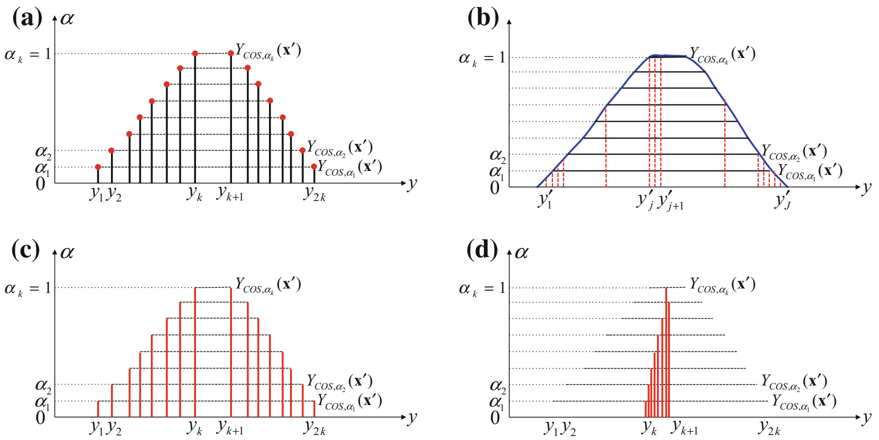


Fig. 11.8 **a** $Y_{\mathrm{WH-COS}}(\mathbf{x}')$ is constructed from its α-cuts (the filled-in *red dots* represent the right-hand side of (11.31); **b** spline approximation to $Y_{\mathrm{WH-COS}}$, $\hat{Y}_{\mathrm{WH-COS}}(y)$, after which the continuous function $\hat{Y}_{\mathrm{WH-COS}}(y)$ is uniformly sampled; **c** spikes of height α are located at the two end-points of each $Y_{\mathrm{WH-COS}}$ in **a**; and, **d** spikes of height α are located at the average of the two end-points of each α-cut of $Y_{\mathrm{WH-COS}}$ (Mendel 2014; © IEEE 2014)

$$y_{\mathrm{WH-1}}(\mathbf{x}') = \frac{\sum_{j=1}^{J} y_j' \hat{Y}_{\mathrm{WH-COS}}(y_j')}{\sum_{j=1}^{J} \hat{Y}_{\mathrm{WH-COS}}(y_j')} \tag{11.40}$$

Clearly, there are a lot of computations needed in order to obtain $y_{\mathrm{WH-1}}(\mathbf{x}')$, which makes it not very practical for real-time applications of a GT2 fuzzy system.

11.7.2 End-Points Defuzzification

In this approach a set of *2k spikes* is created, two spikes located at the end points of each $\alpha/Y_{\mathrm{COS},\alpha}(\mathbf{x}')$, both with amplitude α (see Fig. 11.8c). These spikes are then defuzzified, by computing their COG, as:

$$y_{\mathrm{WH-2}}(\mathbf{x}') = \frac{\sum_{i=1}^{k} \left[\alpha_i y_{l,\alpha_i}^{\mathrm{COS}}(\mathbf{x}') + \alpha_i y_{r,\alpha_i}^{\mathrm{COS}}(\mathbf{x}')\right]}{\sum_{i=1}^{k} 2\alpha_i} \tag{11.41}$$

Note that, in Fig. 11.8c:

$$\begin{cases} y_1 = y_{l,\alpha_1}^{\mathrm{COS}}(\mathbf{x}'),\ y_2 = y_{l,\alpha_2}^{\mathrm{COS}}(\mathbf{x}'), \ldots, y_k = y_{l,\alpha_k}^{\mathrm{COS}}(\mathbf{x}') \\ y_{2k} = y_{r,\alpha_1}^{\mathrm{COS}}(\mathbf{x}'), y_{2k-1} = y_{r,\alpha_2}^{\mathrm{COS}}(\mathbf{x}'), \ldots, y_{k+1} = y_{r,\alpha_k}^{\mathrm{COS}}(\mathbf{x}') \end{cases} \tag{11.42}$$

These values of y are not located uniformly on the y-axis, a fact and not a criticism.

Observe, in (11.41), that $\alpha_i = 0$ makes no contribution to this formula.

11.7.3 Average of End-Points Defuzzification

In this approach one first computes the average value of each $\alpha/Y_{\mathrm{COS},\alpha}(\mathbf{x}')$ and locates a spike at that value with amplitude α (see Fig. 11.8d) after which the k spikes are defuzzified, by computing their COG, as:

$$y_{\mathrm{WH-3}}(\mathbf{x}') = \frac{\sum_{i=1}^{k} \alpha_i \left[\left(y_{l,\alpha_i}^{\mathrm{COS}}(\mathbf{x}') + y_{r,\alpha_i}^{\mathrm{COS}}(\mathbf{x}') \right) \Big/ 2 \right]}{\sum_{i=1}^{k} \alpha_i} \tag{11.43}$$

Observe, in (11.43), that $\alpha_i = 0$ makes no contribution to this formula.

Equation (11.43) was first suggested in Wagner and Hagras (2010), and is a much simpler way to obtain a defuzzified value than by Approximation and Defuzzification; however, it still requires a lot of computation because EIASC or EKM algorithms have to be used in order to compute $y_{l,\alpha_i}^{\mathrm{COS}}(\mathbf{x}')$ and $y_{r,\alpha_i}^{\mathrm{COS}}(\mathbf{x}')$ $(\alpha = \alpha_1, \ldots, \alpha_k)$. Of course, if $2k$ parallel processors are available then there will be a huge speedup in computation time, but this comes at the price of needing a lot of processors. Many in the IT2 fuzzy systems community were unhappy about even needing two such processors for real-time applications, which led to the direct defuzzification approaches that were described in Sect. 9.9. Proposed extensions of those approached to WH GT2 fuzzy systems are given in Sect. 11.10.

Theorem 11.3 (Zhai and Mendel 2012) *End-point and average of end-points defuzzifications give the same results, i.e.,* $y_{\mathrm{WH-2}}(\mathbf{x}') = y_{\mathrm{WH-3}}(\mathbf{x}')$.

Proof Compare (11.41) and (11.43).

Example 11.3 Just as an IT2 fuzzy system should reduce to a T1 fuzzy system when all MF uncertainties disappear, a WH GT2 fuzzy system should reduce to an IT2 fuzzy system when all the GT FSs in the WH GT2 fuzzy system reduce to IT2 FSs. When this occurs, then (Exercise 11.6), e.g.

$$\begin{cases} y_{l,\alpha_1}^{\mathrm{COS}}(\mathbf{x}') = y_{l,\alpha_2}^{\mathrm{COS}}(\mathbf{x}') = \cdots = y_{l,\alpha_k=1}^{\mathrm{COS}}(\mathbf{x}') \equiv y_l^{\mathrm{COS}}(\mathbf{x}') \\ y_{r,\alpha_1}^{\mathrm{COS}}(\mathbf{x}') = y_{r,\alpha_2}^{\mathrm{COS}}(\mathbf{x}') = \cdots = y_{r,\alpha_k=1}^{\mathrm{COS}}(\mathbf{x}') \equiv y_r^{\mathrm{COS}}(\mathbf{x}') \end{cases} \tag{11.44}$$

Substituting (11.44) into (11.43), one finds:

$$\begin{aligned} y_{\text{WH}-3}(\mathbf{x}') &= \frac{\sum_{i=1}^{k} \alpha_i \left[\left(y_{l,\alpha_i}^{\text{COS}}(\mathbf{x}') + y_{r,\alpha_i}^{\text{COS}}(\mathbf{x}') \right) / 2 \right]}{\sum_{i=1}^{k} \alpha_i} \\ &= \frac{\sum_{i=1}^{k} \alpha_i \left[\left(y_l^{\text{COS}}(\mathbf{x}') + y_r^{\text{COS}}(\mathbf{x}') \right) / 2 \right]}{\sum_{i=1}^{k} \alpha_i} \\ &= \frac{y_l^{\text{COS}}(\mathbf{x}') + y_r^{\text{COS}}(\mathbf{x}')}{2} \end{aligned} \tag{11.45}$$

which is in agreement with (9.113).

11.8 Summary

So that the reader can see the forest-from-the-trees, this section summarizes the computations that are needed to implement the following four WH GT2 fuzzy systems that have been covered in Sect. 11.6:

1. Mamdani fuzzy system that uses centroid type-reduction + average of end-points defuzzification
2. Mamdani fuzzy system that uses COS type-reduction + average of end-points defuzzification
3. Unnormalized A2-C0 TSK fuzzy system
4. Normalized A2-C0 TSK fuzzy system

All four of these fuzzy systems share the following common computations:

- Decide on how many α-planes will be used (call this number $k_{\max}$) and which ones they will be, i.e., what $\alpha_1, \alpha_2, \ldots, \alpha_{k_{\max}}$ are.
- Compute[9] and store the $k_{\max}$ α-planes of the p antecedent GT2 FSs and the M consequent GT2 FS $(l = 1, \ldots, M, i = 1, \ldots, p \text{ and } k = 1, \ldots, k_{\max})$:

$$\tilde{F}_{i,\alpha_k}^l = \int_{x_i \in X_i} [a_{i,\alpha_k}^l(x_i), b_{i,\alpha_k}^l(x_i)]/x_i \quad x_i \in X_i \tag{11.46}$$

$$\tilde{G}_{\alpha_k}^l = \int_{y \in Y} [c_{\alpha_k}^l(y), d_{\alpha_k}^l(y)]/y \quad y \in Y \tag{11.47}$$

[9]This step depends on the design choices made for the secondary MFs, and is returned to in Sect. 11.14. In practice, X_i and Y are sampled, and (11.46) and (11.47) are computed for $x_i \in X_{id}$ and $y \in Y_d$. This is also true for later computations.

- For $\mathbf{x} = \mathbf{x}'$ and $k = 1, \ldots, k_{\max}$, compute the level-α_k firing sets, $F^l_{\alpha_k}(\mathbf{x}') \equiv \alpha_k/[\underline{f}^l_{\alpha_k}(\mathbf{x}'), \bar{f}^l_{\alpha_k}(\mathbf{x}')]$, where $\underline{f}^l_{\alpha_k}(\mathbf{x}')$ and $\bar{f}^l_{\alpha_k}(\mathbf{x}')$ are found by setting $\alpha = \alpha_k$ in (11.7) and (11.8), respectively.

11.8.1 WH GT2 Mamdani Fuzzy System that Uses Centroid Type-Reduction + Average of End-Points Defuzzification

All of the following computations are performed for $\mathbf{x} = \mathbf{x}'$, $l = 1, \ldots, M$ and $k = 1, \ldots, k_{\max}$.

- Compute the horizontal-slice (level-α) fired-rule output set ($y \in Y$)

$$\mathrm{FOU}(\tilde{B}^l_{\alpha_k}) = [\underline{\mu}_{\tilde{B}^l_{\alpha_k}}(y|\mathbf{x}'), \bar{\mu}_{\tilde{B}^l_{\alpha_k}}(y|\mathbf{x}')] \tag{11.48}$$

where $\underline{\mu}_{\tilde{B}^l_{\alpha_k}}(y|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}^l_{\alpha_k}}(y|\mathbf{x}')$ are found by setting $\alpha = \alpha_k$ in (11.11) and (11.12), respectively.

- Compute the horizontal-slice aggregated output ($y \in Y$)

$$\mathrm{FOU}(\tilde{B}_{\alpha_k}) = [\underline{\mu}_{\tilde{B}_{\alpha_k}}(y|\mathbf{x}'), \bar{\mu}_{\tilde{B}_{\alpha_k}}(y|\mathbf{x}')] \tag{11.49}$$

  where $\underline{\mu}_{\tilde{B}_{\alpha_k}}(y|\mathbf{x}')$ and $\bar{\mu}_{\tilde{B}_{\alpha_k}}(y|\mathbf{x}')$ are found by setting $\alpha = \alpha_k$ in (11.20) and (11.21), respectively.
- Compute the centroid of $\tilde{B}_{\alpha_k}$ using the EIASC or EKM algorithms, the results being $[c_l(\alpha_k|\mathbf{x}'), c_r(\alpha_k|\mathbf{x}')]$ (Sect. 11.6.1).
- Compute the average of end-points defuzzified output $y_{\mathrm{WH}}(\mathbf{x}')$, whose structure is analogous to (11.43), and is

$$y_{\mathrm{WH}}(\mathbf{x}') = \frac{\sum_{k=1}^{k_{\max}} \alpha_k[([c_l(\alpha_k|\mathbf{x}') + c_r(\alpha_k|\mathbf{x}')])/2]}{\sum_{k=1}^{k_{\max}} \alpha_k} \tag{11.50}$$

11.8.2 WH GT2 Mamdani Fuzzy System that Uses COS Type-Reduction + Average of End-Points Defuzzification

All of the following computations are performed for $\mathbf{x} = \mathbf{x}'$, $l = 1, \ldots, M$ and $k = 1, \ldots, k_{\max}$.

- Compute the centroids of $\tilde{G}^l_{\alpha_k}$ using the EIASC or EKM algorithms, the results being $[c_l(\tilde{G}^l_\alpha), c_r(\tilde{G}^l_\alpha)]$. This only has to be done one time because $c_l(\tilde{G}^l_\alpha)$ and $c_r(\tilde{G}^l_\alpha)$ do not depend on $\mathbf{x}'$.
- Use the EIASC or EKM algorithms to compute $[y^{\mathrm{COS}}_{l,\alpha_k}(\mathbf{x}'), y^{\mathrm{COS}}_{r,\alpha_k}(\mathbf{x}')]$ (as in (11.29) and (11.30) in Sect. 11.6.2).
- Compute the average of end-points defuzzified output, in (11.43), as:

$$y^{\mathrm{COS}}_{\mathrm{WH}}(\mathbf{x}') = \frac{\sum_{k=1}^{k_{\max}} \alpha_k \left[\left(y^{\mathrm{COS}}_{l,\alpha_k}(\mathbf{x}') + y^{\mathrm{COS}}_{r,\alpha_k}(\mathbf{x}') \right) / 2 \right]}{\sum_{k=1}^{k_{\max}} \alpha_k} \tag{11.51}$$

It should be clear that COS type-reduction + average of end-points defuzzification requires many fewer computations than centroid type-reduction + average of end-points defuzzification; hence, if COS type-reduction + defuzzification was favored for an IT2 Mamdani fuzzy system, as it has been by many practitioners, then COS type-reduction + average of end-points defuzzification will be greatly favored for a WH GT2 Mamdani fuzzy system.

11.8.3 Unnormalized A2-C0 WH GT2 TSK Fuzzy System

All of the following computations are performed for $\mathbf{x} = \mathbf{x}'$, $l = 1, \ldots, M$ and $k = 1, \ldots, k_{\max}$.

- Compute $y^U_{\mathrm{TSK},l,\alpha_k}(\mathbf{x}')$ and $y^U_{\mathrm{TSK},r,\alpha_k}(\mathbf{x}')$ by setting $\alpha = \alpha_k$ in (11.33) and (11.34), respectively.
- Compute the average of end-points defuzzified output $y^U_{\mathrm{WH-TSK}}(\mathbf{x}')$, whose structure is analogous to (11.43), and is

$$y^U_{\mathrm{WH-TSK}}(\mathbf{x}') = \frac{\sum_{k=1}^{k_{\max}} \alpha_k \left[\left(y^U_{\mathrm{TSK},l,\alpha_k}(\mathbf{x}') + y^U_{\mathrm{TSK},r,\alpha_k}(\mathbf{x}') \right) \Big/ 2 \right]}{\sum_{k=1}^{k_{\max}} \alpha_k} \tag{11.52}$$

11.8.4 Normalized A2-C0 WH GT2 TSK Fuzzy System

All of the following computations are performed for $\mathbf{x} = \mathbf{x}', l = 1, \ldots, M$ and $k = 1, \ldots, k_{\max}$.

- Compute $y^N_{\text{TSK},l,\alpha_k}(\mathbf{x}')$ and $y^N_{\text{TSK},r,\alpha_k}(\mathbf{x}')$, by setting $\alpha = \alpha_k$ in (11.38) and (11.39), respectively, and using the EKM or EIASC algorithms.
- Compute the average of end-points defuzzified output $y^N_{\text{WH}-\text{TSK}}(\mathbf{x}')$, whose structure is analogous to (11.43), and is:

$$y^N_{\text{WH}-\text{TSK}}(\mathbf{x}') = \frac{\sum_{k=1}^{k_{\max}} \alpha_k \left[\left(y^N_{\text{TSK},l,\alpha_k}(\mathbf{x}') + y^N_{\text{TSK},r,\alpha_k}(\mathbf{x}') \right) \Big/ 2 \right]}{\sum_{k=1}^{k_{\max}} \alpha_k} \tag{11.53}$$

Although the right-hand sides of (11.53) and (11.52) look exactly the same, they are different because of the ways in which $y^N_{\text{TSK},l,\alpha_k}(\mathbf{x}')$, $y^N_{\text{TSK},r,\alpha_k}(\mathbf{x}')$, $y^U_{\text{TSK},l,\alpha_k}(\mathbf{x}')$ and $y^U_{\text{TSK},r,\alpha_k}(\mathbf{x}')$ are computed.

11.9 Comprehensive Example

This is a continuation of the Sect. 9.7 comprehensive example (which is an IT2 version of the comprehensive T1 example that is in Sect. 3.7). The goal here is to illustrate all of the WH GT2 fuzzy set computations so as to once again obtain $y(\mathbf{x}')$ when $x'_1 = -0.3$ and $x'_2 = 0.6$ for the following four-rule fuzzy system [see (9.132)]:

$$\begin{aligned}
&\tilde{R}^1_Z: \text{ IF } x_1 \text{ is } \tilde{F}^1_1 = \tilde{X}_{11} \quad \text{and} \quad x_2 \text{ is } \tilde{F}^1_2 = \tilde{X}_{21}, \text{ THEN } y \text{ is } \tilde{G}^1 \\
&\tilde{R}^2_Z: \text{ IF } x_1 \text{ is } \tilde{F}^2_1 = \tilde{X}_{11} \quad \text{and} \quad x_2 \text{ is } \tilde{F}^2_2 = \tilde{X}_{22}, \text{ THEN } y \text{ is } \tilde{G}^2 \\
&\tilde{R}^3_Z: \text{ IF } x_1 \text{ is } \tilde{F}^3_1 = \tilde{X}_{12} \quad \text{and} \quad x_2 \text{ is } \tilde{F}^3_2 = \tilde{X}_{21}, \text{ THEN } y \text{ is } \tilde{G}^3 \\
&\tilde{R}^4_Z: \text{ IF } x_1 \text{ is } \tilde{F}^4_1 = \tilde{X}_{12} \quad \text{and} \quad x_2 \text{ is } \tilde{F}^4_2 = \tilde{X}_{22}, \text{ THEN } y \text{ is } \tilde{G}^4
\end{aligned} \tag{11.54}$$

For illustrative purposes, this example is for a WH GT2 Mamdani fuzzy system that uses COS type-reduction + average of end-points defuzzification, in which $\alpha_k = 0.25, 0.5, 0.75, 1$, so that in (11.50) $k_{\max} = 4$. Even though the $\alpha_k = 0$ results are not used to compute $y(\mathbf{x}')$, they are also given below so as to connect the GT2 fuzzy system results with the Sect. 9.7 IT2 fuzzy system results.

Although the antecedent T2 FSs in (11.54) are GT2 FSs, for the purposes of this example (which is for singleton fuzzification) it is not necessary to specify their entire 3D MFs, because it is only their secondary MFs (vertical slices) at $x'_1 = -0.3$ and $x'_2 = 0.6$ that are needed in the computations. For illustrative purposes, triangle secondary MFs are used whose apex factor $w = 0.6$ (see Example 6.21), and whose bases are attached to the Fig. 11.9a FOUs. Because the equations for the α-cuts of

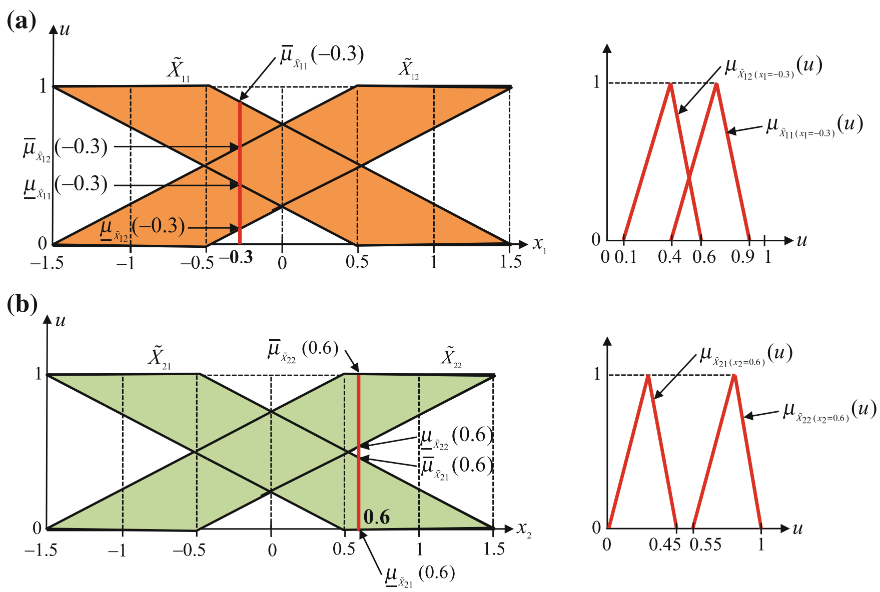


Fig. 11.9 FOUs (the *left* portions of Fig. 11.9a, b are the same as Fig. 9.13a, b, and are repeated here for the convenience of the reader) of the GT2 FSs and their secondary MFs when $x_1' = -0.3$ and $x_2' = 0.6$. **a** x_1 quantities (the apexes of the two secondary MF triangles are located at 0.4 and 0.7), and **b** x_2 quantities (the apexes of the two secondary MF triangles are located at 0.27 and 0.82)

these triangle MFs, which are given in (6.54), are used here, they are also repeated next for the convenience of the reader, but only for $w = 0.6$, and using the GT2 FS notations in (11.54) $(i, j = 1, 2;\ k = 1, \ldots, 4,\ \text{ and }\ x = x_1 \text{ or } x_2)$:

$$\begin{cases} \tilde{X}_{ij}(x)_{\alpha_k} = [a^{ij}_{\alpha_k}(x), b^{ij}_{\alpha_k}(x)] \\ a^{ij}_{\alpha_k}(x) = \underline{\mu}_{\tilde{X}_{ij}}(x) + 0.6[\bar{\mu}_{\tilde{X}_{ij}}(x) - \underline{\mu}_{\tilde{X}_{ij}}(x)]\alpha_k \\ b^{ij}_{\alpha_k}(x) = \bar{\mu}_{\tilde{X}_{ij}}(x) - 0.4[\bar{\mu}_{\tilde{X}_{ij}}(x) - \underline{\mu}_{\tilde{X}_{ij}}(x)]\alpha_k \end{cases} \tag{11.55}$$

α-cuts of these triangle MFs, for $\alpha_k = 0.25, 0.5, 0.75, 1$, are provided in Table 11.1. Equation (9.133) was used in the computation of these α-cuts, e.g., because (see Fig. 11.9a) $\underline{\mu}_{\tilde{X}_{11}}(-0.3) = 0.4$ and $\bar{\mu}_{\tilde{X}_{11}}(-0.3) = 0.9$, so that $a^{11}_{\alpha_k}(-0.3) = 0.4 + 0.3\alpha_k$ and $b^{11}_{\alpha_k}(-0.3) = 0.9 - 0.2\alpha_k$.

Because COS type-reduction is used in this example, one does not need the entire 3D MFs of the four consequent GT2 FSs, $\tilde{G}^l$. Instead, as was done in Sect. 9.7, only the centroids of the consequents at $\alpha_k = 0.25, 0.5, 0.75, 1$ need to specified. The approach that was taken to do this is:

1. Start with the centroids that are given in Table 9.1, and use them as the $\alpha = 0$ centroids of $\tilde{G}^l$ $(l = 1, 2, 3, 4)$.

Table 11.1 α-cuts of the antecedent secondary MFs, two of which occur at $x_1' = -0.3$ and two of which occur at $x_2' = 0.6$

	$\tilde{X}_{11}(x_1 = -0.3)_{\alpha_k} \equiv [a_{\alpha_k}^{11}(-0.3), b_{\alpha_k}^{11}(-0.3)]$		$\tilde{X}_{12}(x_1 = -0.3)_{\alpha_k} \equiv [a_{\alpha_k}^{12}(-0.3), b_{\alpha_k}^{12}(-0.3)]$	
α_k	$a_{\alpha_k}^{11}(-0.3) = 0.4 + 0.3\alpha_k$	$b_{\alpha_k}^{11}(-0.3) = 0.9 - 0.2\alpha_k$	$a_{\alpha_k}^{12}(-0.3) = 0.1 + 0.3\alpha_k$	$b_{\alpha_k}^{12}(-0.3) = 0.6 - 0.2\alpha_k$
0	0.400	0.900	0.100	0.600
0.25	0.475	0.850	0.175	0.550
0.50	0.550	0.800	0.250	0.500
0.75	0.625	0.750	0.325	0.450
1	0.700	0.700	0.400	0.400
	$\tilde{X}_{21}(x_2 = 0.6)_{\alpha_k} \equiv [a_{\alpha_k}^{21}(0.6), b_{\alpha_k}^{21}(0.6)]$		$\tilde{X}_{22}(x_2 = 0.6)_{\alpha_k} \equiv [a_{\alpha_k}^{22}(0.6), b_{\alpha_k}^{22}(0.6)]$	
α_k	$a_{\alpha_k}^{21}(0.6) = 0.27\alpha_k$	$b_{\alpha_k}^{21}(0.6) = 0.45 - 0.18\alpha_k$	$a_{\alpha_k}^{22}(0.6) = 0.55 + 0.27\alpha_k$	$b_{\alpha_k}^{22}(0.6) = 1 - 0.18\alpha_k$
0	0	0.450	0.550	1
0.25	0.068	0.405	0.618	0.955
0.50	0.135	0.360	0.685	0.910
0.75	0.203	0.315	0.753	0.865
1	0.270	0.270	0.820	0.820

$\alpha_k = 0.25, 0.5, 0.75, 1$; the $\alpha_k = 0$ values are in agreement with (9.133)

2. Assume that the secondary MFs for each $\tilde{G}^l$ is a triangle and use observation P6, made in Sect. 8.4.1, that for such secondary MFs the shape of the centroid (a T1 FS) is itself *triangle-looking*.
3. Use the formulas for the α-cuts of a triangle that are given in Table 2.3 in which m^l is the apex of the triangle, m_1^l is the location of the left vertex and m_2^l is the location of the right vertex.
4. Choose m^l arbitrarily for each of the consequent GT2 FSs, and fix (Table 2.3) m_1^l and m_2^l at their $\alpha = 0$ values from Step 1.
5. Compute the approximate centroid α-cuts using the following formulas (from Table 2.3) $(l, k = 1, \ldots, 4)$:

$$\begin{cases} c_l(\tilde{G}_\alpha^l) \approx m_1^l + (m^l - m_1^l)\alpha_k \\ c_r(\tilde{G}_\alpha^l) \approx m_2^l - (m_2^l - m^l)\alpha_k \end{cases} \tag{11.56}$$

Numerical values are summarized in Table 11.2, which is the counterpart to Table 9.1. It is the left and right end-points of these α-cuts that are used in the next set of computations

Formulas for the level-α firing intervals for the four rules, computed using the product t-norm in (11.7) and (11.8) and using the notations for this example, are:

$$\begin{cases} [\underline{f}_{\alpha_k}^1, \bar{f}_{\alpha_k}^1] = [a_{\alpha_k}^{11}(-0.3) \times a_{\alpha_k}^{21}(0.6),\ b_{\alpha_k}^{11}(-0.3) \times b_{\alpha_k}^{21}(0.6)] \\ [\underline{f}_{\alpha_k}^2, \bar{f}_{\alpha_k}^2] = [a_{\alpha_k}^{11}(-0.3) \times a_{\alpha_k}^{22}(0.6),\ b_{\alpha_k}^{11}(-0.3) \times b_{\alpha_k}^{22}(0.6)] \\ [\underline{f}_{\alpha_k}^3, \bar{f}_{\alpha_k}^3] = [a_{\alpha_k}^{12}(-0.3) \times a_{\alpha_k}^{21}(0.6),\ b_{\alpha_k}^{12}(-0.3) \times b_{\alpha_k}^{21}(0.6)] \\ [\underline{f}_{\alpha_k}^4, \bar{f}_{\alpha_k}^4] = [a_{\alpha_k}^{12}(-0.3) \times a_{\alpha_k}^{22}(0.6),\ b_{\alpha_k}^{12}(-0.3) \times b_{\alpha_k}^{22}(0.6)] \end{cases} \tag{11.57}$$

These are analogous to the formulas that are in Table 9.2. Table 11.3 provides numerical values for these level α firing intervals for $\alpha_k = 0.25, 0.5, 0.75, 1$. They are not shown for $\alpha = 0$ because the $\alpha = 0$ plane does not contribute anything to the final defuzzified value. The numerical values used to compute the left- and right-ends of the level-α firing intervals are taken from Table 11.1, and $[c_l(\tilde{G}_{\alpha_k}^l), c_r(\tilde{G}_{\alpha_k}^l)]$ are taken from Table 11.2.

Next COS type-reduction is performed for $\alpha_k = 0.25, 0.5, 0.75, 1$. Because there are only four rules, this is done, for the reader's edification, by using exhaustive search, i.e., by computing [see (11.29) and (11.30)][10]:

[10]For notational simplification, y_{l,α_k}^{COS} and y_{l,α_k}^{COS} are shortened to y_{l,α_k} and y_{r,α_k}, respectively.

Table 11.2 α-cuts of the centroids raised to level α of the four consequent GT2 FSs

Rule 1: $C_{\tilde{G}^1_{\alpha_k}} = \alpha_k/[c_l(\tilde{G}^1_{\alpha_k}), c_r(\tilde{G}^1_{\alpha_k})]$					
α_k	m^1	m^1_1	m^1_2	$c_l(\tilde{G}^1_{\alpha_k})$	$c_r(\tilde{G}^1_{\alpha_k})$
0	−0.93	−1	−0.90	−1.000	−0.900
0.25	−0.93	−1	−0.90	−0.983	−0.908
0.50	−0.93	−1	−0.90	−0.965	−0.915
0.75	−0.93	−1	−0.90	−0.948	−0.923
1	−0.93	−1	−0.90	−0.930	−0.930
Rule 2: $C_{\tilde{G}^2_{\alpha_k}} = \alpha_k/[c_l(\tilde{G}^2_{\alpha_k}), c_r(\tilde{G}^2_{\alpha_k})]$					
α_k	m^2	m^2_1	m^2_2	$c_l(\tilde{G}^2_{\alpha_k})$	$c_r(\tilde{G}^2_{\alpha_k})$
0	−0.48	−0.60	−0.40	−0.600	−0.400
0.25	−0.48	−0.60	−0.40	−0.570	−0.420
0.50	−0.48	−0.60	−0.40	−0.540	−0.440
0.75	−0.48	−0.60	−0.40	−0.510	−0.460
1	−0.48	−0.60	−0.40	−0.480	−0.480
Rule 3: $C_{\tilde{G}^3_{\alpha_k}} = \alpha_k/[c_l(\tilde{G}^3_{\alpha_k}), c_r(\tilde{G}^3_{\alpha_k})]$					
α_k	m^3	m^3_1	m^3_2	$c_l(\tilde{G}^3_{\alpha_k})$	$c_r(\tilde{G}^3_{\alpha_k})$
0	0.50	0.40	0.60	0.400	0.600
0.25	0.50	0.40	0.60	0.425	0.575
0.50	0.50	0.40	0.60	0.450	0.550
0.75	0.50	0.40	0.60	0.475	0.525
1	0.50	0.40	0.60	0.500	0.500
Rule 4: $C_{\tilde{G}^4_{\alpha_k}} = \alpha_k/[c_l(\tilde{G}^4_{\alpha_k}), c_r(\tilde{G}^4_{\alpha_k})]$					
α_k	m^4	m^4_1	m^4_2	$c_l(\tilde{G}^4_{\alpha_k})$	$c_r(\tilde{G}^4_{\alpha_k})$
0	0.96	0.90	1	0.400	0.600
0.25	0.96	0.90	1	0.915	0.990
0.50	0.96	0.90	1	0.930	0.980
0.75	0.96	0.90	1	0.945	0.970
1	0.96	0.90	1	0.960	0.960

m^l were chosen arbitrarily; $\alpha_k = 0.25, 0.5, 0.75, 1$; and, $\alpha_k = 0$ values are in agreement with Table 9.1 (Step 1)

$$\begin{aligned}
y^{(1)}_{l,\alpha_k}(\mathbf{x}') &= \frac{\underline{f}^1_{\alpha_k} \times c_l(\tilde{G}^1_{\alpha_k}) + \underline{f}^2_{\alpha_k} \times c_l(\tilde{G}^2_{\alpha_k}) + \underline{f}^3_{\alpha_k} \times c_l(\tilde{G}^3_{\alpha_k}) + \underline{f}^4_{\alpha_k} \times c_l(\tilde{G}^4_{\alpha_k})}{\underline{f}^1_{\alpha_k} + \underline{f}^2_{\alpha_k} + \underline{f}^3_{\alpha_k} + \underline{f}^4_{\alpha_k}} \\
y^{(2)}_{l,\alpha_k}(\mathbf{x}') &= \frac{\bar{f}^1_{\alpha_k} \times c_l(\tilde{G}^1_{\alpha_k}) + \underline{f}^2_{\alpha_k} \times c_l(\tilde{G}^2_{\alpha_k}) + \underline{f}^3_{\alpha_k} \times c_l(\tilde{G}^3_{\alpha_k}) + \underline{f}^4_{\alpha_k} \times c_l(\tilde{G}^4_{\alpha_k})}{\bar{f}^1_{\alpha_k} + \underline{f}^2_{\alpha_k} + \underline{f}^3_{\alpha_k} + \underline{f}^4_{\alpha_k}} \\
y^{(3)}_{l,\alpha_k}(\mathbf{x}') &= \frac{\bar{f}^1_{\alpha_k} \times c_l(\tilde{G}^1_{\alpha_k}) + \bar{f}^2_{\alpha_k} \times c_l(\tilde{G}^2_{\alpha_k}) + \underline{f}^3_{\alpha_k} \times c_l(\tilde{G}^3_{\alpha_k}) + \underline{f}^4_{\alpha_k} \times c_l(\tilde{G}^4_{\alpha_k})}{\bar{f}^1_{\alpha_k} + \bar{f}^2_{\alpha_k} + \underline{f}^3_{\alpha_k} + \underline{f}^4_{\alpha_k}} \\
y^{(4)}_{l,\alpha_k}(\mathbf{x}') &= \frac{\bar{f}^1_{\alpha_k} \times c_l(\tilde{G}^1_{\alpha_k}) + \bar{f}^2_{\alpha_k} \times c_l(\tilde{G}^2_{\alpha_k}) + \bar{f}^3_{\alpha_k} \times c_l(\tilde{G}^3_{\alpha_k}) + \underline{f}^4_{\alpha_k} \times c_l(\tilde{G}^4_{\alpha_k})}{\bar{f}^1_{\alpha_k} + \bar{f}^2_{\alpha_k} + \bar{f}^3_{\alpha_k} + \underline{f}^4_{\alpha_k}}
\end{aligned} \tag{11.58}$$

and

$$
\begin{aligned}
y^{(1)}_{r,\alpha_k}(\mathbf{x}') &= \frac{\bar{f}^1_{\alpha_k} \times c_r(\tilde{G}^1_{\alpha_k}) + \bar{f}^2_{\alpha_k} \times c_r(\tilde{G}^2_{\alpha_k}) + \bar{f}^3_{\alpha_k} \times c_r(\tilde{G}^3_{\alpha_k}) + \bar{f}^4_{\alpha_k} \times c_r(\tilde{G}^4_{\alpha_k})}{\bar{f}^1_{\alpha_k} + \bar{f}^2_{\alpha_k} + \bar{f}^3_{\alpha_k} + \bar{f}^4_{\alpha_k}} \\
y^{(2)}_{r,\alpha_k}(\mathbf{x}') &= \frac{\underline{f}^1_{\alpha_k} \times c_r(\tilde{G}^1_{\alpha_k}) + \bar{f}^2_{\alpha_k} \times c_r(\tilde{G}^2_{\alpha_k}) + \bar{f}^3_{\alpha_k} \times c_r(\tilde{G}^3_{\alpha_k}) + \bar{f}^4_{\alpha_k} \times c_r(\tilde{G}^4_{\alpha_k})}{\underline{f}^1_{\alpha_k} + \bar{f}^2_{\alpha_k} + \bar{f}^3_{\alpha_k} + \bar{f}^4_{\alpha_k}} \\
y^{(3)}_{r,\alpha_k}(\mathbf{x}') &= \frac{\underline{f}^1_{\alpha_k} \times c_r(\tilde{G}^1_{\alpha_k}) + \underline{f}^2_{\alpha_k} \times c_r(\tilde{G}^2_{\alpha_k}) + \bar{f}^3_{\alpha_k} \times c_r(\tilde{G}^3_{\alpha_k}) + \bar{f}^4_{\alpha_k} \times c_r(\tilde{G}^4_{\alpha_k})}{\underline{f}^1_{\alpha_k} + \underline{f}^2_{\alpha_k} + \bar{f}^3_{\alpha_k} + \bar{f}^4_{\alpha_k}} \\
y^{(4)}_{r,\alpha_k}(\mathbf{x}') &= \frac{\underline{f}^1_{\alpha_k} \times c_r(\tilde{G}^1_{\alpha_k}) + \underline{f}^2_{\alpha_k} \times c_r(\tilde{G}^2_{\alpha_k}) + \underline{f}^3_{\alpha_k} \times c_r(\tilde{G}^3_{\alpha_k}) + \bar{f}^4_{\alpha_k} \times c_r(\tilde{G}^4_{\alpha_k})}{\underline{f}^1_{\alpha_k} + \underline{f}^2_{\alpha_k} + \underline{f}^3_{\alpha_k} + \bar{f}^4_{\alpha_k}}
\end{aligned}
\tag{11.59}
$$

The level α firing intervals and centroid end-points on each α-plane are taken from Table 11.3. A careful examination of the left (right) end-points of the consequent

Table 11.3 Level α firing intervals and rule consequent centroid intervals $[c_l(\tilde{G}^l_{\alpha_k}), c_r(\tilde{G}^l_{\alpha_k})]$ for $\alpha_k = 0.25, 0.5, 0.75, 1$ and $l = 1, \ldots, 4$

	$\alpha_1 = 0.25$	
Rule number (l)	Level α firing interval-$[\underline{f}^l_{\alpha_k}, \bar{f}^l_{\alpha_k}]$	$[c_l(\tilde{G}^l_{\alpha_k}), c_r(\tilde{G}^l_{\alpha_k})]$
$\tilde{R}^1_Z$	$[0.475 \times 0.068, 0.85 \times 0.405] = [0.032, 0.344]$	$[-0.983, -0.908]$
$\tilde{R}^2_Z$	$[0.475 \times 0.618, 0.85 \times 0.955] = [0.293, 0.812]$	$[-0.57, -0.42]$
$\tilde{R}^3_Z$	$[0.175 \times 0.068, 0.55 \times 0.405] = [0.012, 0.223]$	$[0.425, 0.575]$
$\tilde{R}^4_Z$	$[0.175 \times 0.618, 0.55 \times 0.955] = [0.108, 0.525]$	$[0.915, 0.99]$
	$\alpha_2 = 0.5$	
Rule number (l)	Level α firing interval-$[\underline{f}^l_{\alpha_k}, \bar{f}^l_{\alpha_k}]$	$[c_l(\tilde{G}^l_{\alpha_k}), c_r(\tilde{G}^l_{\alpha_k})]$
$\tilde{R}^1_Z$	$[0.55 \times 0.135, 0.8 \times 0.36] = [0.074, 0.288]$	$[-0.965, -0.915]$
$\tilde{R}^2_Z$	$[0.55 \times 0.685, 0.8 \times 0.9] = [0.377, 0.728]$	$[-0.54, -0.44]$
$\tilde{R}^3_Z$	$[0.25 \times 0.135, 0.5 \times 0.36] = [0.034, 0.18]$	$[0.45, 0.55]$
$\tilde{R}^4_Z$	$[0.25 \times 0.685, 0.5 \times 0.91] = [0.171, 0.455]$	$[0.93, 0.98]$
	$\alpha_3 = 0.75$	
Rule number (l)	Level α firing interval-$[\underline{f}^l_{\alpha_k}, \bar{f}^l_{\alpha_k}]$	$[c_l(\tilde{G}^l_{\alpha_k}), c_r(\tilde{G}^l_{\alpha_k})]$
$\tilde{R}^1_Z$	$[0.625 \times 0.203, 0.75 \times 0.315] = [0.127, 0.236]$	$[-0.948, -0.923]$
$\tilde{R}^2_Z$	$[0.625 \times 0.753, 0.75 \times 0.865] = [0.470, 0.649]$	$[-0.51, -0.46]$
$\tilde{R}^3_Z$	$[0.325 \times 0.203, 0.45 \times 0.315] = [0.066, 0.142]$	$[0.475, 0.525]$
$\tilde{R}^4_Z$	$[0.325 \times 0.753, 0.45 \times 0.865] = [0.245, 0.389]$	$[0.945, 0.97]$
	$\alpha_4 = 1$	
Rule number (l)	Level α firing interval-$[\underline{f}^l_{\alpha_k}, \bar{f}^l_{\alpha_k}]$	$[c_l(\tilde{G}^l_{\alpha_k}), c_r(\tilde{G}^l_{\alpha_k})]$
$\tilde{R}^1_Z$	$[0.7 \times 0.27, 0.7 \times 0.27] = 0.189$	-0.93
$\tilde{R}^2_Z$	$[0.7 \times 0.82, 0.7 \times 0.82] = 0.574$	-0.48
$\tilde{R}^3_Z$	$[0.4 \times 0.27, 0.4 \times 0.27] = 0.108$	0.5
$\tilde{R}^4_Z$	$[0.4 \times 0.82, 0.4 \times 0.82] = 0.328$	0.96

centroid end-points on each α-plane shows that they are already in an increasing order; hence, no reordering of them is needed.

Table 11.4 summarizes the four iterations for the left and right end-points of $Y_{\mathrm{COS},\alpha}(\mathbf{x}')$ for $\alpha_k = 0.25, 0.5, 0.75, 1$. For this particular choice of $\mathbf{x}'$ it happens that the switch points for the left (right)-end of $Y_{\mathrm{COS},\alpha}(\mathbf{x}')$ all occur at the same value for these four values of α_k. This would not necessarily be so for other choices of $\mathbf{x}'$. The solutions for $\alpha_k = 1$ are all the same because when triangle secondary MFs are used then the horizontal slice at $\alpha_k = 1$ is a T1 FS.

Finally, the defuzzified value $y_{\mathrm{WH}}^{\mathrm{COS}}(\mathbf{x}')$ is obtained using (11.51), as:

$$y_{\mathrm{WH}}^{\mathrm{COS}}(\mathbf{x}') = \frac{\sum_{k=1}^{4} \alpha_k[(y_{l,\alpha_k}^{\mathrm{COS}}(\mathbf{x}') + y_{r,\alpha_k}^{\mathrm{COS}}(\mathbf{x}'))/2]}{\sum_{k=1}^{4} \alpha_k} \tag{11.60}$$

All of the numbers that are needed on the right-hand side of (11.60) are in Table 11.4, so that:

$$\begin{aligned} y_{WH}^{COS}(\mathbf{x}') &= \frac{0.25 \times (-0.546 + 0.462) + 0.5 \times (-0.407 + 0.287) + 0.75 \times (-0.244 + 0.105) + 1 \times (-0.069)}{0.25 + 0.50 + 0.75 + 1} \\ &= -0.065 \end{aligned} \tag{11.61}$$

Table 11.4 COS type-reduction iterations for $\alpha_k = 0.25, 0.5, 0.75, 1$

$\alpha_1 = 0.25$								
$y_{l,0.25}^{(1)}(\mathbf{x}')$	$y_{l,0.25}^{(2)}(\mathbf{x}')$	$y_{l,0.25}^{(3)}(\mathbf{x}')$	$y_{l,0.25}^{(4)}(\mathbf{x}')$	$y_{r,0.25}^{(1)}(\mathbf{x}')$	$y_{r,0.25}^{(2)}(\mathbf{x}')$	$y_{r,0.25}^{(3)}(\mathbf{x}')$	$y_{r,0.25}^{(4)}(\mathbf{x}')$	$Y_{COS,0.25}(\mathbf{x}')$
–0.212	–0.530	–0.546	–0.408	–0.003	0.175	0.462	0.175	[–0.546,0.462]
$\alpha_2 = 0.5$								
$y_{l,0.50}^{(1)}(\mathbf{x}')$	$y_{l,0.50}^{(2)}(\mathbf{x}')$	$y_{l,0.50}^{(3)}(\mathbf{x}')$	$y_{l,0.50}^{(4)}(\mathbf{x}')$	$y_{r,0.50}^{(1)}(\mathbf{x}')$	$y_{r,0.50}^{(2)}(\mathbf{x}')$	$y_{r,0.50}^{(3)}(\mathbf{x}')$	$y_{r,0.50}^{(4)}(\mathbf{x}')$	$Y_{COS,0.50}(\mathbf{x}')$
–0.153	–0.353	–0.407	–0.315	–0.024	0.109	0.287	0.246	[–0.407,0.287]
$\alpha_3 = 0.75$								
$y_{l,0.75}^{(1)}(\mathbf{x}')$	$y_{l,0.75}^{(2)}(\mathbf{x}')$	$y_{l,0.75}^{(3)}(\mathbf{x}')$	$y_{l,0.75}^{(4)}(\mathbf{x}')$	$y_{r,0.75}^{(1)}(\mathbf{x}')$	$y_{r,0.75}^{(2)}(\mathbf{x}')$	$y_{r,0.75}^{(3)}(\mathbf{x}')$	$y_{r,0.75}^{(4)}(\mathbf{x}')$	$Y_{COS,0.75}(\mathbf{x}')$
–0.107	–0.197	–0.244	–0.201	–0.046	0.028	0.105	0.075	[–0.244,0.105]
$\alpha_4 = 1$								
$y_{l,1}^{(1)}(\mathbf{x}')$	$y_{l,1}^{(2)}(\mathbf{x}')$	$y_{l,1}^{(3)}(\mathbf{x}')$	$y_{l,1}^{(4)}(\mathbf{x}')$	$y_{r,1}^{(1)}(\mathbf{x}')$	$y_{r,1}^{(2)}(\mathbf{x}')$	$y_{r,1}^{(3)}(\mathbf{x}')$	$y_{r,1}^{(4)}(\mathbf{x}')$	$Y_{COS,1}(\mathbf{x}')$
–0.069	–0.069	–0.069	–0.069	–0.069	–0.069	–0.069	–0.069	–0.069

Shaded numbers are the winners

Comparing this value with the defuzzified value of $y_{\text{COS}}(\mathbf{x}') = -0.047$ that was obtained for the Mamdani IT2 fuzzy system, in Sect. 9.7 [Eq. (9.136)], observe that they are significantly different, indicating that significant differences can occur at the output of a WH GT2 fuzzy system as compared to the output of an IT2 fuzzy system.

Section 9.7 also carried out detail computations for both unnormalized and normalized A2-C0 IT2 TSK fuzzy systems. Exercise 11.8 asks the reader to do this for both unnormalized and normalized WH A2-C0 GT2 TSK fuzzy systems.

11.10 Direct Defuzzification

Referring to Fig. 11.1b, direct defuzzification means not having to perform type-reduction. The ground-rule for direct defuzzification is still the following fundamental design requirement: *If all sources of MF uncertainty disappear then a GT2 fuzzy system must reduce to a T1 fuzzy system*. This is interpreted here to also mean that *when all GT2 FSs reduce to IT2 FSs, a GT2 fuzzy system must reduce to an IT2 fuzzy system*. Type-reduction + defuzzification is not the only way to satisfy the fundamental design requirement. Direct defuzzification can also do this.

Although the main motivation for direct defuzzification is to avoid the iterative computations needed to perform type-reduction, another motivation for it is to obtain a formula for the crisp output of a GT2 fuzzy system that can be used in some sort of mathematical analyses, e.g., stability, robustness.

The direct defuzzification formulas described below are *conjectured* extensions of the Nie-Tan (NT) and Biglarbegian–Melek–Mendel (BMM) direct defuzzification formulas, that are described in Sects. 9.9.1 and 9.9.2, respectively,[11] to their WH formulations. As of the year 2017, neither extension has been applied yet to any real-world application (to the best knowledge of this author); however, since their IT2 fuzzy system counterparts have been, there is no reason to believe that they will not be. Consequently, the descriptions of these extensions that are given next are provided to expedite this.

11.10.1 Proposed WH-NT Direct Defuzzification

Section 9.9.1 presents three NT formulas for direct defuzzification in an IT2 fuzzy system. Equations (9.155) and (9.156) both begin with the combined fired-rule

[11]The extension of approximate type-reduction + defuzzification (Wu-Mendel uncertainty bounds), that is given in Sect. 9.8, is left to the reader as an exercise (Exercise 11.12) because its formulas are much more complicated than the ones for the extensions of the NT and BMM formulas.

output set $\tilde{B}$ and then defuzzify $\tilde{B}$ directly by computing the COS of the average of its lower and upper MFs. Equations (9.155) uses sampled values of $\tilde{B}$, whereas (9.156) assumes that mathematical formulas are available for the lower and upper MFs of $\tilde{B}$, and uses integrals. Equation (9.157), on the other hand, replaces height type-reduction with an averaging technique, and uses the average of the end-points of the firing interval. The horizontal-slice versions of those equations are ($\alpha \in [0, 1]$):

$$\begin{aligned} y_{\mathrm{NT},\alpha}(\mathbf{x}') &= \mathrm{COG}\left\{\frac{1}{2}\left[\underline{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}') + \bar{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}')\right]\right\} \\ &= \frac{\sum_{i=1}^{N} y_i \left[\underline{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}') + \bar{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}')\right]}{\sum_{i=1}^{N} \left[\underline{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}') + \bar{\mu}_{\tilde{B}_\alpha}(y_i|\mathbf{x}')\right]} \end{aligned} \tag{11.62}$$

$$y_{\mathrm{NT},\alpha}(\mathbf{x}') = \frac{\int_{a_\alpha}^{b_\alpha} y\left[\underline{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}') + \bar{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}')\right]dy}{\int_{a_\alpha}^{b_\alpha} \left[\underline{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}') + \bar{\mu}_{\tilde{B}_\alpha}(y|\mathbf{x}')\right]dy} \tag{11.63}$$

$$y_{\mathrm{NT},\alpha}(\mathbf{x}') = \frac{\sum_{l=1}^{M} \bar{y}_\alpha^l \left[\underline{f}_\alpha^l(\mathbf{x}') + \bar{f}_\alpha^l(\mathbf{x}')\right]}{\sum_{l=1}^{M} \left[\underline{f}_\alpha^l(\mathbf{x}') + \bar{f}_\alpha^l(\mathbf{x}')\right]} \tag{11.64}$$

Each $y_{\mathrm{NT},\alpha}(\mathbf{x}')$ is a crisp number that is located along the y-axis with amplitude α. After $y_{\mathrm{NT},\alpha}(\mathbf{x}')$ are computed for $\alpha = \alpha_k = \alpha_1, \ldots, \alpha_{k_{\max}}$, $y_{\mathrm{WH-NT}}(\mathbf{x}')$ can be computed as the COG of those spikes, as:

$$y_{\mathrm{WH-NT}}(\mathbf{x}') = \frac{\sum_{k=1}^{k_{\max}} \alpha_k y_{\mathrm{NT},\alpha_k}(\mathbf{x}')}{\sum_{k=1}^{k_{\max}} \alpha_k} \tag{11.65}$$

Focusing on $y_{\mathrm{NT},\alpha}(\mathbf{x}')$ in (11.64), the steps to computing $y_{\mathrm{WH-NT}}(\mathbf{x}')$ in (11.65) are (Exercises 11.13 and 11.14 ask the reader to set up the computational steps beginning with $y_{\mathrm{NT},\alpha}(\mathbf{x}')$ in (11.62) and (11.63), respectively):

1. Decide on how many α-planes will be used (call this number $k_{\max}$) and which ones they will be, i.e., what $\alpha_1, \alpha_2, \ldots, \alpha_{k_{\max}}$ are.
2. Compute[12] the $k_{\max}$ α-cuts of the p antecedent vertical slices at $x_i = x_i'$, namely $(l = 1, \ldots, M,\ \ i = 1, \ldots, p$ and $k = 1, \ldots, k_{\max})[a_{i,\alpha_k}^l(x_i'), b_{i,\alpha_k}^l(x_i')]$ (see footnote 12).
3. For $\mathbf{x} = \mathbf{x}'$, compute the level-α_k firing sets ($k = 1, \ldots, k_{\max}$ and $l = 1, \ldots, M$) $F_{\alpha_k}^l(\mathbf{x}') \equiv \alpha_k/[\underline{f}_{\alpha_k}^l(\mathbf{x}'), \bar{f}_{\alpha_k}^l(\mathbf{x}')]$, where $\underline{f}_{\alpha_k}^l(\mathbf{x}')$ and $\bar{f}_{\alpha_k}^l(\mathbf{x}')$ are found by setting $\alpha = \alpha_k$ in (11.7) and (11.8), respectively.

[12]This step depends on the design choices made for the secondary MFs, and is returned to in Sect. 11.14.

4. Treat the $\bar{y}^l_{\alpha_k}$ in (11.64) as $k_{\max}M$ design parameters that will be optimized during the design of a GT2 WH -NT fuzzy system [if $\bar{y}^l_{\alpha_k}$ are chosen to be the same (this is a design choice) for $\alpha_k = \alpha_1, \ldots, \alpha_{k_{\max}}$, then there will only be M design parameters].
5. Compute $y_{\mathrm{NT},\alpha}(\mathbf{x}')$ in (11.64).
6. Compute $y_{\mathrm{WH-NT}}(\mathbf{x}')$ in (11.65).

11.10.2 Proposed WH-BMM Direct Defuzzification

Section 9.9.2 presents the BMM formula for direct defuzzification in an IT2 fuzzy system. The horizontal-slice version of this formula is:

$$y_{\mathrm{BMM}_\alpha}(\mathbf{x}') = m_\alpha \frac{\sum_{l=1}^{M} y_{l,\alpha} \underline{f}^l_\alpha(\mathbf{x}')}{\sum_{l=1}^{M} \underline{f}^l_\alpha(\mathbf{x}')} + n_\alpha \frac{\sum_{l=1}^{M} y_{l,\alpha} \bar{f}^l_\alpha(\mathbf{x}')}{\sum_{l=1}^{M} \bar{f}^l_\alpha(\mathbf{x}')} \tag{11.66}$$

Each $y_{\mathrm{BMM}_\alpha}(\mathbf{x}')$ is a crisp number that is located along the y-axis with amplitude α. After $y_{\mathrm{BMM}_\alpha}(\mathbf{x}')$ are computed for $\alpha = \alpha_k = \alpha_1, \ldots, \alpha_{k_{\max}}$, $y_{\mathrm{WH-BMM}}(\mathbf{x}')$ can be computed as the COG of those spikes, as:

$$y_{\mathrm{WH-BMM}}(\mathbf{x}') = \frac{\sum_{k=1}^{k_{\max}} \alpha_k y_{\mathrm{BMM}_{\alpha_k}}(\mathbf{x}')}{\sum_{k=1}^{k_{\max}} \alpha_k} \tag{11.67}$$

The steps to computing $y_{\mathrm{WH-BMM}}(\mathbf{x}')$ in (11.67) are:

1. Decide on how many α-planes will be used (call this number $k_{\max}$) and which ones they will be, i.e., what $\alpha_1, \alpha_2, \ldots, \alpha_{k_{\max}}$ are.
2. Compute the $k_{\max}$ α-cuts of the p antecedent vertical slices at $x_i = x_i'$, namely $(l = 1, \ldots, M, i = 1, \ldots, p$ and $k = 1, \ldots, k_{\max})$ $[a^l_{i,\alpha_k}(x_i'), b^l_{i,\alpha_k}(x_i')]$ (see footnote 12).
3. For $\mathbf{x} = \mathbf{x}'$, compute the level-α_k firing sets $(k = 1, \ldots, k_{\max}$ and $l = 1, \ldots, M)$ $F^l_{\alpha_k}(\mathbf{x}') \equiv \alpha_k / [\underline{f}^l_{\alpha_k}(\mathbf{x}'), \bar{f}^l_{\alpha_k}(\mathbf{x}')]$, where $\underline{f}^l_{\alpha_k}(\mathbf{x}')$ and $\bar{f}^l_{\alpha_k}(\mathbf{x}')$ are found by setting $\alpha = \alpha_k$ in (11.7) and (11.8), respectively.
4. Treat $y_{l.\alpha}$, m_α and n_α as $k_{\max}(M+2)$ design parameters that will be optimized during the design of a GT2 WH -BMM fuzzy system [if $m_\alpha = m$ and $n_\alpha = n$ for all α, then there will only be $k_{\max}M + 2$ design parameters; however, if the same $y_{l,\alpha}$ is not used in both terms of (11.67) there will be even more design parameters].
5. Compute $y_{\mathrm{BMM}_\alpha}(\mathbf{x}')$ in (11.66).
6. Compute $y_{\mathrm{WH-BMM}}(\mathbf{x}')$ in (11.67).

11.11 Comprehensive Example Continued

This is a continuation of the Sect. 11.9 comprehensive example, so it is advisable for the reader to review that section. Here, the data from that section are used, and $y_{\mathrm{NT}}(\mathbf{x}')$ is computed when $\mathbf{x}' = \mathrm{col}(-0.3, 0.6)$. The computation of $y_{\mathrm{BMM}}(\mathbf{x}')$ is left as an exercise for the reader because it requires more quantities to be specified ahead of time than does $y_{\mathrm{NT}}(\mathbf{x}')$ (Exercise 11.16).

Following the six-step procedure given in Sect. 11.10.1:

1. $k_{\max}$ is chosen to be 4, and $\alpha_1 = 0.25$, $\alpha_2 = 0.5$, $\alpha_3 = 0.75$ and $\alpha_4 = 1$.
2. The four α-cuts of the antecedent vertical slices are already in Table 11.1.
3. The level-α_k firing intervals are in already Table 11.3.
4. Four choices are made for the parameters $\bar{y}^l_{\alpha_k}$ in (11.64), so that the effects of those choices can be observed on the final computed value of $y_{\mathrm{NT}}(\mathbf{x}')$ $(l, k = 1, \ldots, 4)$:
   a. $\bar{y}^l_{\alpha_k} = [c_l(\tilde{G}^l_{\alpha_k}) + c_r(\tilde{G}^l_{\alpha_k})]/2$ ($c_l(\tilde{G}^l_{\alpha_k})$ and $c_r(\tilde{G}^l_{\alpha_k})$ are in the last column Table 11.3).
   b. $\bar{y}^l_{\alpha_k} = c_l(\tilde{G}^l_{\alpha_k})$ ($c_l(\tilde{G}^l_{\alpha_k})$ is in the last column Table 11.3).
   c. $\bar{y}^l_{\alpha_k} = c_r(\tilde{G}^l_{\alpha_k})$ ($c_r(\tilde{G}^l_{\alpha_k})$ is in the last column Table 11.3).
   d. $\bar{y}^l_{\alpha_k} = g^l(\mathbf{x}')$ ($g^l(\mathbf{x}')$ is in the last column of Table 9.3).
5. $y_{\mathrm{NT},\alpha}(\mathbf{x}')$ is computed using (11.64), and is tabulated in Table 11.5 for the four choices of $\bar{y}^l_{\alpha_k}$ and for the four values of α_k.
6. $y_{\mathrm{WH-NT}}(\mathbf{x}')$ is computed using (11.65) and is tabulated in the last row of Table 11.5.

The following observations are based on the results that are in Table 11.5:

- $y_{\mathrm{WH-NT}}(\mathbf{x}')$ can be quite different depending upon the choice made for the $\bar{y}^l_{\alpha}$, which suggests that the tuning of these parameters will affect the numerical value of $y_{\mathrm{WH-NT}}(\mathbf{x}')$, which is good.
- When $\bar{y}^l_{\alpha}$ are chosen as $[c_l(\tilde{G}^l_{\alpha_k}) + c_r(\tilde{G}^l_{\alpha_k})]/2$, the numerical values of the four $y_{\mathrm{NT},\alpha}(\mathbf{x}')$ are clustered close to each other, and $y_{\mathrm{WH-NT}}(\mathbf{x}') = -0.072$ is very

Table 11.5 Computations of $y_{\mathrm{NT},\alpha}(\mathbf{x}')$ and $y_{\mathrm{WH-NT}}(\mathbf{x}')$ for four choices of $\bar{y}^l_{\alpha}$

NT	Choices made for $\bar{y}^l_{\alpha}$			
$y_{\mathrm{NT},\alpha}(\mathbf{x}')$	$\bar{y}^l_{\alpha_k} = \left[c_l(\tilde{G}^l_{\alpha_k}) + c_r(\tilde{G}^l_{\alpha_k})\right]/2$	$\bar{y}^l_{\alpha_k} = c_l(\tilde{G}^l_{\alpha_k})$	$\bar{y}^l_{\alpha_k} = c_r(\tilde{G}^l_{\alpha_k})$	$\bar{y}^l_{\alpha_k} = g^l(\mathbf{x}')$
$y_{\mathrm{NT},0.25}(\mathbf{x}')$	−0.075	−0.134	−0.016	3.231
$y_{\mathrm{NT},0.50}(\mathbf{x}')$	−0.077	−0.116	−0.037	3.229
$y_{\mathrm{NT},0.75}(\mathbf{x}')$	−0.074	−0.093	−0.054	3.229
$y_{\mathrm{NT},1}(\mathbf{x}')$	−0.069	−0.069	−0.069	3.230
$y_{WH-NT}(\mathbf{x}')$	−0.072	−0.092	−0.132	3.230

close to those values; this value is also quite close to the output of the IT2 NT fuzzy system, which is given in (9.169), as -0.077.

- When $\bar{y}^l_\alpha$ are chosen as $g^l(\mathbf{x}')$, which means that $\bar{y}^l_\alpha$ is the same for all four values of α_k, the numerical values of the four $y_{\mathrm{NT},\alpha}(\mathbf{x}')$ are clustered very close to each other, there is just about no difference between $y_{\mathrm{WH-NT}}(\mathbf{x}')$ and any one of the $y_{\mathrm{NT},\alpha}(\mathbf{x}')$, and the value of 3.23 for $y_{\mathrm{WH-NT}}(\mathbf{x}')$ is the same as the output of the IT2 NT fuzzy system [see (9.168)].

The last two results and observations raise the question: Why are $y_{\mathrm{WH-NT}}(\mathbf{x}')$ and $y_{\mathrm{NT}}(\mathbf{x}')$ so alike? Two thoughts come to mind:

1. The results in this example are only for one choice of $\mathbf{x}'$, so it is hasty to jump to a general conclusion based only this one point.
2. The secondary MFs in Fig. 11.9 are close to symmetrical ($w = 0.6$ and symmetry occurs for $w = 0.5$), and perhaps it is this closeness to symmetry that is causing $y_{\mathrm{WH-NT}}(\mathbf{x}')$ to be close to $y_{\mathrm{NT}}(\mathbf{x}')$. In practice, each apex factor of a triangle secondary MS is a design parameter that would be optimized during the design process, and it is not very likely that each apex factor would wind up being close to 0.5 (see Exercise 11.15).

11.12 GT2 Fuzzy Basis Functions

Just as it was useful to describe the output of a specific IT2 fuzzy system as an IT2 fuzzy basis function (FBF) expansion (Sect. 9.12), it is also useful to do the same for the output of a specific WH GT2 fuzzy system. Examples 9.13–9.16 provide the IT2 FBF expansion for four specific IT2 fuzzy systems. Here, it is explained how to use the results in those examples to obtain GT2 FBF expansions for the WH GT2 fuzzy systems that are described in Sects. 11.8.2, 11.8.3, 11.10.1 and 11.10.2.

Example 11.4 (**WH GT2 Mamdani fuzzy system that uses COS type-reduction + average of end-points defuzzification**): The starting point for a *GT2 FBF expansion* for this system is (11.51), which can be reorganized as:

$$\begin{aligned} y_{\mathrm{WH}}^{\mathrm{COS}}(\mathbf{x}) &= \frac{1}{2}\frac{\sum_{k=1}^{k_{\max}} \alpha_k y_{l,\alpha_k}^{\mathrm{COS}}(\mathbf{x})}{\sum_{k=1}^{k_{\max}} \alpha_k} + \frac{1}{2}\frac{\sum_{k=1}^{k_{\max}} \alpha_k y_{r,\alpha_k}^{\mathrm{COS}}(\mathbf{x})}{\sum_{k=1}^{k_{\max}} \alpha_k} \\ &= \frac{1}{2}\sum_{k=1}^{k_{\max}} w_k y_{l,\alpha_k}^{\mathrm{COS}}(\mathbf{x}) + \frac{1}{2}\sum_{k=1}^{k_{\max}} w_k y_{r,\alpha_k}^{COS}(\mathbf{x}) \end{aligned} \tag{11.68}$$

$$w_k \equiv \frac{\alpha_k}{\sum_{i=1}^{k_{\max}} \alpha_i} \tag{11.69}$$

As in Example 9.13, in which (8.18) and (8.20) were used in (9.113) instead of (9.111) and (9.112), the same is done here. In fact, the approach here is to convert

$y_l^{\text{COS}}(\mathbf{x})$ and $y_r^{\text{COS}}(\mathbf{x})$, that are given in (9.172), as well as (9.173)–(9.176), into their following horizontal-slice counterparts:

$$y_{l,\alpha_k}^{\text{COS}}(\mathbf{x}) = \sum_{i=1}^{M} c_l(\tilde{G}_{\alpha_k}^i)\phi_{l,\alpha_k}^i(\mathbf{x}) \tag{11.70}$$

$$y_{r,\alpha_k}^{\text{COS}}(\mathbf{x}) = \sum_{i=1}^{M} c_r(\tilde{G}_{\alpha_k}^i)\phi_{r,\alpha_k}^i(\mathbf{x}) \tag{11.71}$$

$$\phi_{l,\alpha_k}^i(\mathbf{x}) = \frac{\delta_{l,\alpha_k}^i \bar{f}_{\alpha_k}^i(\mathbf{x}) + (1-\delta_{l,\alpha_k}^i)\underline{f}_{\alpha_k}^i(\mathbf{x})}{\sum_{i=1}^{M}\left[\delta_{l,\alpha_k}^i \bar{f}_{\alpha_k}^i(\mathbf{x}) + (1-\delta_{l,\alpha_k}^i)\underline{f}_{\alpha_k}^i(\mathbf{x})\right]} \tag{11.72}$$

$$\phi_{r,\alpha_k}^i(\mathbf{x}) = \frac{\delta_{r,\alpha_k}^i \bar{f}_{\alpha_k}^i(\mathbf{x}) + (1-\delta_{r,\alpha_k}^i)\underline{f}_{\alpha_k}^i(\mathbf{x})}{\sum_{i=1}^{M}\left[\delta_{r,\alpha_k}^i \bar{f}_{\alpha_k}^i(\mathbf{x}) + (1-\delta_{r,\alpha_k}^i)\underline{f}_{\alpha_k}^i(\mathbf{x})\right]} \tag{11.73}$$

$$\delta_{l,\alpha_k}^i = \begin{cases} 1 & c_l(\tilde{G}_{\alpha_k}^i) \le y_{l,\alpha_k}^{\text{COS}}(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{11.74}$$

$$\delta_{r,\alpha_k}^i = \begin{cases} 1 & c_r(\tilde{G}_{\alpha_k}^i) \ge y_{r,\alpha_k}^{\text{COS}}(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{11.75}$$

Observe, in (11.68), (11.70) and (11.71), that each horizontal slice has $2M$ IT2 α-level FBFs, where $\{\phi_{l,\alpha_k}^i(\mathbf{x})\}_{i=1}^M$ and $\{\phi_{r,\alpha_k}^i(\mathbf{x})\}_{i=1}^M$ (the *GT2 FBFs*) both depend on a mixture of the lower and upper firing intervals on that horizontal slice. Consequently, in (11.68) $y_{\text{WH}}^{\text{COS}}(\mathbf{x})$ has $2k_{\max}M$ GT2 FBFs. The fact that there is a factor of $k_{\max}$ more FBFs for a WH GT2 fuzzy system as compared to an IT2 fuzzy system suggests that improved performance should be possible for a WH GT2 fuzzy system over an IT2 fuzzy system.

The discussions that are in the second paragraph below (9.176) about not actually being able to use the IT2 FBF expansions to compute with apply as well for the WH GT2 fuzzy system, since each horizontal slice has its own IT2 FBF expansion. $y_{\text{WH}}^{\text{COS}}(\mathbf{x}')$ is computed as explained in Sect. 11.8.2.

Example 11.5 **(Unnormalized A2-C0 WH GT2 TSK fuzzy system)**: The starting point for a *GT2 FBF expansion* for this system is (11.52) which can be reorganized as:

$$\begin{aligned} y_{\text{WH-TSK}}^U(\mathbf{x}) &= \frac{1}{2}\frac{\sum_{k=1}^{k_{\max}} \alpha_k y_{TSK,l,\alpha_k}^U(\mathbf{x})}{\sum_{k=1}^{k_{\max}} \alpha_k} + \frac{1}{2}\frac{\sum_{k=1}^{k_{\max}} \alpha_k y_{\text{TSK},r,\alpha_k}^U(\mathbf{x})}{\sum_{k=1}^{k_{\max}} \alpha_k} \\ &= \frac{1}{2}\sum_{k=1}^{k_{\max}} w_k y_{\text{TSK},l,\alpha_k}^U(\mathbf{x}) + \frac{1}{2}\sum_{k=1}^{k_{\max}} w_k y_{\text{TSK},r,\alpha_k}^U(\mathbf{x}) \end{aligned} \tag{11.76}$$

where w_k is defined in (11.69) and $y^U_{\mathrm{TSK},l,\alpha_k}(\mathbf{x})$ and $y^U_{\mathrm{TSK},r,\alpha_k}$ are given in (11.33) and (11.34), respectively. As in Example 9.14, it is assumed here that the consequent $g^i(\mathbf{x})$ of each GT2 TSK rule is $(x_0 \equiv 1)$:

$$g^i(\mathbf{x}) = c^i_0 + c^i_1 x_1 + c^i_2 x_2 + \cdots + c^i_p x_p = \sum_{j=0}^{p} c^l_j x_j \tag{11.77}$$

Substituting (11.33), (11.34), and (11.77) into (11.76), one obtains:

$$\begin{aligned} y^U_{\mathrm{WH-TSK}}(\mathbf{x}) &= \sum_{k=1}^{k_{\max}} w_k \left[\sum_{l=1}^{M} g^l(\mathbf{x}) \left(\frac{\underline{f}^l_{\alpha_k}(\mathbf{x}) + \bar{f}^l_{\alpha_k}(\mathbf{x})}{2} \right) \right] \\ &= \sum_{k=1}^{k_{\max}} w_k \left[\sum_{l=1}^{M} \sum_{j=0}^{p} c^l_j x_j \left(\frac{\underline{f}^l_{\alpha_k}(\mathbf{x}) + \bar{f}^l_{\alpha_k}(\mathbf{x})}{2} \right) \right] \end{aligned} \tag{11.78}$$

Equation (11.78) can be expressed, using horizontal-slice versions of (9.178) and (9.179), as:

$$y^U_{\mathrm{WH-TSK}}(\mathbf{x}) = \sum_{k=1}^{k_{\max}} w_k y^U_{\mathrm{WH-TSK},\alpha_k}(\mathbf{x}) \tag{11.79}$$

$$y^U_{\mathrm{WH-TSK},\alpha_k}(\mathbf{x}) = \sum_{l=1}^{M} \sum_{j=0}^{p} c^l_j \phi^l_{j,\alpha_k}(\mathbf{x}) \tag{11.80}$$

$$\phi^l_{j,\alpha_k}(\mathbf{x}) = \left[\frac{\underline{f}^l_{\alpha_k}(\mathbf{x}) + \bar{f}^l_{\alpha_k}(\mathbf{x})}{2} \right] \times x_j \tag{11.81}$$

Observe in (11.80) that each horizontal slice has $M(p+1)$ *GT2 FBFs*, where $\{\phi^i_{j,\alpha_k}(\mathbf{x}),\ j = 1, \ldots, p$ and $l = 1, \ldots, M\}$ depend on a mixture of the lower and upper firing intervals on that horizontal slice. Consequently, $y^U_{\mathrm{WH-TSK}}(\mathbf{x})$ in (11.79) has $M(p+1)k_{\max}$ GT2 FBFs. The fact that there is a factor of $k_{\max}$ more FBFs for an unnormalized A2-C0 WH GT2 TSK fuzzy system as compared to an unnormalized A2-C0 IT2 TSK fuzzy system again suggests that improved performance should be possible for the former fuzzy system over the latter fuzzy system.

Example 11.6 (**Proposed WH-NT direct defuzzification**): The starting point for a *GT2 FBF expansion* for this system is (11.65) which can be reexpressed as:

$$y_{\mathrm{WH-NT}}(\mathbf{x}) = \sum_{k=1}^{k_{\max}} w_k y_{\mathrm{NT},\alpha_k}(\mathbf{x}) \tag{11.82}$$

where w_k is in (11.69) and $y_{\mathrm{NT},\alpha_k}(\mathbf{x})$ is given by[13] (11.63) or (11.64), depending upon the NT formulation. For example, for (11.64), (11.82) can be expressed by using the horizontal-slice versions of (9.180) and (9.181), as:

$$y_{\mathrm{WH-NT}}(\mathbf{x}) = \sum_{k=1}^{k_{\max}} w_k \underbrace{\sum\nolimits_{l=1}^{M} \bar{y}^l_{\alpha_k} \phi_{l,\alpha_k}(\mathbf{x})}_{y_{\mathrm{NT},\alpha_k}(\mathbf{x})} \tag{11.83}$$

$$\phi_{l,\alpha_k}(\mathbf{x}) = \frac{\underline{f}^l_{\alpha_k}(\mathbf{x}) + \bar{f}^l_{\alpha_k}(\mathbf{x})}{\sum_{j=1}^{M}\left[\underline{f}^j_{\alpha_k}(\mathbf{x}) + \bar{f}^j_{\alpha_k}(\mathbf{x})\right]} \tag{11.84}$$

Observe from (11.83) that each horizontal slice $y_{\mathrm{NT},\alpha_k}(\mathbf{x})$ has *M GT2 FBFs*, where $\phi_{l,\alpha_k}(\mathbf{x})$ depends on a mixture of the lower and upper firing intervals on that horizontal slice. Consequently, $y_{\mathrm{WH-NT}}(\mathbf{x})$ in (11.83) has $Mk_{\max}$ GT2 FBFs. The fact that there is a factor of $k_{\max}$ more FBFs for a WH-NT GT2 fuzzy system as compared to a NT IT2 fuzzy system again suggests that improved performance should be possible for a WH-NT GT2 fuzzy system over a NT IT2 fuzzy system.

Example 11.7 (**Proposed WH-BMM direct defuzzification**): The starting point for a *GT2 FBF expansion* for this system is (11.67) which can be reexpressed as:

$$y_{\mathrm{WH-BMM}}(\mathbf{x}) = \sum_{k=1}^{k_{\max}} w_k y_{\mathrm{WH-BMM},\alpha_k}(\mathbf{x}) \tag{11.85}$$

where w_k is defined in (11.69) and $y_{\mathrm{BMM},\alpha_k}(\mathbf{x})$ is given by (11.66). Using the horizontal-slice versions of (9.183)–(9.185), (11.85) can be reexpressed as:

$$y_{\mathrm{WH-BMM}}(\mathbf{x}) = \sum_{k=1}^{k_{\max}} w_k \sum_{i=1}^{M} y_{i,\alpha_k} \phi^i_{l,\alpha_k}(\mathbf{x}) + \sum_{k=1}^{k_{\max}} w_k \sum_{i=1}^{M} y_{i,\alpha_k} \phi^i_{r,\alpha_k}(\mathbf{x}) \tag{11.86}$$

$$\phi^i_{l,\alpha_k}(\mathbf{x}) = \frac{m_\alpha \underline{f}^i_{\alpha_k}(\mathbf{x})}{\sum_{i=1}^{M} \underline{f}^i_{\alpha_k}(\mathbf{x})} \tag{11.87}$$

$$\phi^i_{r,\alpha_k}(\mathbf{x}) = \frac{n_\alpha \bar{f}^i_{\alpha_k}(\mathbf{x})}{\sum_{i=1}^{M} \bar{f}^i_{\alpha_k}(\mathbf{x})} \tag{11.88}$$

Observe from (11.85) and (11.86) that each horizontal slice $y_{\mathrm{WH-BMM},\alpha_k}(\mathbf{x})$ has $2M$ *GT2 FBFs*, where $\{\phi^i_{l,\alpha_k}(\mathbf{x})\}_{i=1}^{M}$ depend on the lower firing intervals on that

[13] Equation (11.62) is excluded here because (see, also footnote 14 in Chap. 3) this author does not feel that the number of FBFs should depend on the discretization, N, of the primary variable.

horizontal slice, and $\{\phi^i_{r,\alpha_k}(\mathbf{x})\}_{i=1}^M$ depend on the upper firing intervals on that horizontal slice. Consequently, $y_{\mathrm{WH-BMM}}(\mathbf{x})$ in (11.86) has $2Mk_{\max}$ GT2 FBFs. Again, the fact that there is a factor of $k_{\max}$ more FBFs for a BMM GT2 fuzzy system as compared to BMM IT2 fuzzy system suggests that improved performance should be possible for a BMM GT2 fuzzy system over a BMM IT2 fuzzy system.

Exercises 11.17–11.22 ask the reader to establish GT2 FBF expansions for other WH GT2 fuzzy systems.

11.13 Remarks and Insights

Because GT2 fuzzy systems are in their infancy, as of the writing of this book, all remarks and insights about them are collected here in one section, unlike Sect. 9.13, which has seven subsections.

Regarding a *layered architecture interpretation for a WH GT2 fuzzy system*, Fig. 11.10 is a layered version of Fig. 11.2. Since the fuzzifier is the same for all horizontal slices, it is removed from them. Within each of the $k_{\max}$ horizontal-slice fuzzy systems, there will be more layers, depending upon the specific WH GT2 fuzzy system that is used. For example, for the WH GT2 Mamdani fuzzy system that uses COS type-reduction + average of end-points defuzzification, there will be two more layers, one for computing the firing intervals at level-k and one for computing the COS type-reduced set at level-k (see Fig. 9.16 and Exercise 11.23). It should be clear from Fig. 11.10 that much parallel processing is possible in a WH GT2 fuzzy system.

Regarding *fundamental differences between IT2 and WH GT2 fuzzy systems*, for those WH GT2 fuzzy systems that include type-reduction, adaptiveness[14] and novelty can now occur on each of the $k_{\max}$ horizontal-slice fuzzy systems. Recall the earlier examples about partitions. Having horizontal-slice first-and second-order rule partitions (and horizontal-slice novelty rule partitions, when type-reduction is used) is another difference between a WH GT2 fuzzy system and an IT2 fuzzy system. Finally, a WH GT2 fuzzy system has a factor of $k_{\max}$ more basis functions than does an IT2 fuzzy system.

Regarding *universal approximation for WH GT2 fuzzy systems*, as of the writing of this book there are no publications about this. I suspect that each of them is a universal approximator, but that it will be very tedious to prove this. And, after such a proof occurs, it will not really help much in designing a WH GT2 fuzzy system because of the existence-only nature of universal approximation theorems.

[14] Adaptiveness and novelty are defined in Sect. 9.13.2.

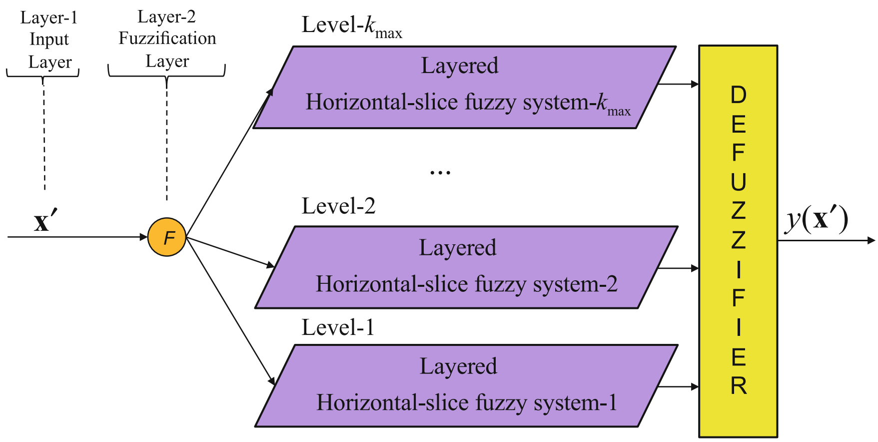


Fig. 11.10 A layered version of Fig. 11.2. Additional layers go inside each of the "Layered Horizontal-slice fuzzy system-k" blocks, depending upon the kind of WH GT2 fuzzy system

Regarding *continuity of WH GT2 fuzzy systems*, as of the writing of this book there are no publications about this. I suspect that by applying the continuity results that are in Sect. 9.13.4 to each of the k_{max} horizontal-slice fuzzy systems, continuity can be achieved for any of the WH GT2 fuzzy systems. This may impose some new restrictions on the choices that can be made for the secondary MFs.

Regarding *rule explosion and some ways to control it for WH GT2 fuzzy systems*, as of the writing of this book there are no publications about this. Whether or not secondary MFs can contribute to further reducing the number of rules as compared to rule reduction by using IT2 FSs remains to be studied.

Regarding *rule interpretability*, as of the writing of this book there are no publications about this. See the discussions about this in Sects. 9.13.6 and 3.9.5. Until one is able to collect word data from a group of subjects and map that data into a GT2 FS, it is not clear how a GT2 FS (unlike an IT2 FS) can be associated with semantics-based interpretability. How to achieve GT2 FS models for words remains to be studied.

Regarding *historical notes*, see Mendel (2010). Although a GT2 fuzzy system was proposed in Karnik et al. (1999) no attention was paid to such fuzzy systems until the emergence of the horizontal-slice representation of a GT2 FS in Liu (2008) and Wagner and Hagras (2008) because of the inherent complexities of computing the join and meet operations for GT2 FSs. Instead, attention was focused entirely on IT2 fuzzy systems. The horizontal-slice representation of a GT2 FS and the WH approach to GT2 fuzzy systems has allowed everything that has been developed for an IT2 fuzzy system to be used by a WH GT2 fuzzy system and has added great value to learning about IT2 fuzzy systems.

11.14 Designing WH GT2 Fuzzy Systems

From the detailed discussions given about the five or four elements that comprise the Fig. 11.1a or 11.1b GT2 fuzzy systems, it should be clear that there are many possibilities to choose from. Table 11.6 enumerates them. Not included in that table is which input variables should be used, because this is application dependent. As was stated in Sects. 4.1 and 10.1:

> For some applications the choice of which variables to use is easy because of historical precedence (e.g., for fuzzy logic control one would choose the system's states). For others, the choice may be very challenging because there may be no historical precedence (e.g., for time-series forecasting one would choose delayed versions of the main variable, however, how many delays to choose is usually unknown and may need to be varied). In the latter situation, it is best to seek the advice of experts, or, if none are available, to create more than one fuzzy system in order to see which choices lead to a best design.

Although portions of Table 11.6 are the same as Table 10.1, some of its rows do not appear in Table 10.1 or are not as self-explanatory as they were for Table 10.1, and so some explanations are provided for them next.

1. *Type-reduction* + *D*: For a WH GT2 fuzzy system, there are separate discussions for type-reduction and defuzzification in Sects. 11.6 and 11.7, respectively, because type-reduction has to be performed for each horizontal-slice fuzzy system (see Fig. 11.10), whereas defuzzification is performed across all of the horizontal-slice fuzzy system. Although Sect. 11.7 describes three kinds of defuzzifiers (two of which give the same numerical results), it is assumed that when "D" is used in Table 11.6 it refers to average of end-points defuzzification and not to approximation and defuzzification, because the latter is computationally too costly. This is why type-reduction + D are not separated in this Table.
2. *Kind of FOU*: Recall that Sect. 6.7.4 concluded:

   (a) *When more than a few horizontal slices are used, the horizontal-slice representation of a GT2 FS may not be a useful representation for the optimal design of a GT2 fuzzy system, but if only a small number of horizontal slices are used, and a squishing technique*[15] *is used, then the horizontal-slice representation of a GT2 FS is a useful representation for the optimal design of a GT2 fuzzy system*. So, when a horizontal-slice representation is used, then in Table 11.6, make the following choices: (i) Kind of FOU[16]–Choose "Specify it"; (ii) Kind of UMFs–Choose "NA"; (iii) Kind of LMFs–Choose "NA"; (iv) FOU parameters–Choose "Pre-specified" or "Optimized"; (v) Kind of Secondary MF–Choose "other," and put "Result of squishing" into it; (vi) Secondary MF parameters–Choose "NA"; (vii) Squishing

[15] A *squishing technique* leads to a nested set of horizontal slices. Two such techniques are described in Sect. 6.7.4. See also Exercises 6.18–6.20.

[16] Recall that the FOU of a GT2 FS is its $\alpha = 0$ horizontal slice.

Table 11.6 Choices that need to be made to specify or design a WH GT2 fuzzy system

Choices for	Kind of WH GT2 fuzzy system	
• Rules	WH-Mamdani	WH-TSK (A2-C0, A2-C1)
• Fuzzifier	Singleton; T1 non-singleton; IT2 non-singleton	Singleton; T1 non-singleton; IT2 non-singleton
• t-norm	Minimum; Product	Minimum; Product
• Combining fired rule output sets	Union; None	NA
• Type-reduction + D	Centroid; Height; COS; NA	Unnormalized; Normalized
• Direct D	WH-NT; WH-BMM; NA	NA
• Number of input variables	p	p
• Number and names of terms per input	$\{X_{1j}\}_{j=1}^{Q_1}$ $\{X_{2j}\}_{j=1}^{Q_2}$ $\cdots$ $\{X_{pj}\}_{j=1}^{Q_p}$	$\{X_{1j}\}_{j=1}^{Q_1}$ $\{X_{2j}\}_{j=1}^{Q_2}$ $\cdots$ $\{X_{pj}\}_{j=1}^{Q_p}$
• Number and names of terms for, or structure of, output	$\{Y_j\}_{j=1}^{Q_y}$	$\{g^l(\mathbf{x})\}_{l=1}^{M}$
• Number of rules	M	M
• Kind of FOU[a]	Specify it; NA	Specify it; NA
• Kind of UMFs	Triangle; Trapezoid; Other; NA	Triangle; Trapezoid; Other; NA
• Kind of LMFs	Triangle; Trapezoid; Other; NA	Triangle; Trapezoid; Other; NA
• FOU parameters	Pre-specified; Optimized	Pre-specified; Optimized
• Kind of secondary MF	Triangle; Trapezoid; Other	Triangle; Trapezoid; Other
• Secondary MF parameters	Pre-specified; Optimized; NA	Pre-specified; Optimized; NA
• Squishing parameters	Pre-specified; Optimized; NA	Pre-specified; Optimized; NA
• Number of horizontal slices	k_{max}	k_{max}

TR denotes type-reduction and D denotes defuzzification

[a]If "Kind of FOU" is specified, then "Kind of UMFs" and "Kind of LMFs " must be specified as "NA." If "Kind of FOU" is "NA" then both "Kind of LMF" and "Kind of UMF" must be specified

Parameters–Choose "pre-specified " or "optimized"; and, (viii) Number of horizontal slices–Enter a small value for $k_{\max}$.

(b) *The vertical-slice representation of a GT2 FS is a very parsimonious representation for the optimal design of a GT2 fuzzy system.* A way to parameterize the vertical-slice representation is: (i) Parameterize its FOU exactly as one presently parameterizes the FOU of an IT2 FS; (ii) Parameterize the secondary MFs by choosing a fairly simple function that introduces only one new parameter; and, (iii) If performance is not acceptable then use secondary MFs that can be described by two parameters, etc. Because the secondary MFs are vertical slices, they are always anchored on the already-parameterized FOU.

3. *Kind of secondary MF*: Triangles[17] and trapezoids are shown in Table 11.6 because they can be described by a small number of parameters. Gaussian MFs are also parsimonious. Note that if a particle-based method is used to design the fuzzy systems, then one must choose trapezoidal secondary MFs, so that an IT2 fuzzy set particle can be embedded in the GT2 fuzzy set particle (see Example 11.10).

Two examples are given next to illustrate the use of Table 11.6 when the vertical-slice representation is chosen.

Example 11.8 Table 11.7 is for a WH GT2 Mamdani fuzzy system which uses singleton fuzzification, product t-norm, center-of-sets type-reduction + D (so fired-rule output sets are not combined, and direct D is not applicable), four input variables, three terms for each variable all of which are named *low*, *moderate,* and *high*, four output terms named *none to very little*, *some*, *a moderate amount* and *a lot*, 81 rules (3^4), all FOUs are like the ones in Table 6.1 (so that the kinds of LMFs and UMFs are NA), FOU parameters will be optimized (the FOUs for the leftmost and rightmost words will be left and right shoulder FOUs, respectively), all secondary MFs are symmetrical trapezoids (as in Example 6.22) whose apex(w)-parameters will be optimized, squishing parameters are NA, and $k_{\max} = 3$.

Example 11.9 Table 11.9 is for a A2-C0 WH GT2 TSK fuzzy system which uses singleton fuzzification, minimum t-norm, normalized type-reduction + D (so that combining fired-rule out sets is NA), three input variables, three terms for each variable all of which are named *light*, *moderate* and *heavy*, a consequent function for each rule that is a crisp linear combination of the three input variables, 27 rules (3^3), normal triangle UMFs and LMFs [the apex of the left (right)-sided right triangle for *light* (*heavy*) will occur at 0 ($\max x_3$)] all of whose free parameters are prespecified, all secondary MFs are symmetrical trapezoids (as in Example 6.22)

[17]For a different kind of design that also uses triangle secondary MFs, see Starczewski (2009). Since the advent of the horizontal-slice decomposition, it is arguably no longer necessary to use the approximate approach to type-reduction that is advocated in this paper. Instead, one only needs to design a WH GT2 fuzzy system that uses a few horizontal slices.

Table 11.7 Example of choices (shaded) made to specify or design a WH GT2 Mamdani fuzzy system

Choices for	Kind of WH GT2 fuzzy system	
• Rules	WH-Mamdani	WH-TSK: A2-C0; A2-C1
• Fuzzifier	Singleton; T1 non-singleton; IT2 non-singleton	Singleton; T1 non-singleton; IT2 non-singleton
• t-norm	Minimum; Product	Minimum; Product
• Combining fired rule output sets	Union; None	NA
• Type-reduction +D	Centroid; Height; COS; NA	Unnormalized; Normalized
• Direct D	WH-NT; WH-BMM; NA	NA
• Number of input variables	4	p
• Number and names of terms per input	*low*; *moderate*; *high*	$\{X_{1j}\}_{j=1}^{Q_1}$ $\{X_{2j}\}_{j=1}^{Q_2}$ $\cdots$ $\{X_{pj}\}_{j=1}^{Q_p}$
• Number and names of terms for, or structure of, output	*none to very little*; *some*; *a moderate amount*; *a lot*	$\{g^l(\mathbf{x})\}_{l=1}^{M}$
• Number of rules	81	M
• Kind of FOU[a]	Specify it; NA	Specify it; NA
• Kind of UMFs	Triangle; Trapezoid; Other; NA	Triangle; Trapezoid; Other; NA
• Kind of LMFs	Triangle; Trapezoid; Other; NA	Triangle; Trapezoid; Other; NA
• FOU parameters	Pre-specified; Optimized	Pre-specified; Optimized
• Kind of secondary MF	Triangle; Trapezoid; Other	Triangle; Trapezoid; Other
• Secondary MF parameters	Pre-specified; Optimized; NA	Pre-specified; Optimized; NA
• Squishing parameters	Pre-specified; Optimized; NA	Pre-specified; Optimized; NA
• Number of horizontal slices	3	$k_{\max}$

TR denotes type-reduction and D denotes defuzzification

[a]If "Kind of FOU" is specified, then "Kind of UMFs" and "Kind of LMFs " must be specified as "NA." If "Kind of FOU" is "NA" then both "Kind of LMF" and "Kind of UMF" must be specified.

whose apex(w)-parameters will be optimized, squishing parameters are NA, and $k_{\max} = 5$.

A design problem for a singleton WH GT2 fuzzy system (e.g., any of the ones that are in Table 11.6) is:

Given N input–output numerical data *training* pairs, $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \ldots, (\mathbf{x}^{(N)} : y^{(N)})$, where $\mathbf{x}$ is the vector input and y is the scalar output of a singleton WH GT2 fuzzy system, completely specify such a fuzzy system using the training data.

At the time of the preparation of this book none of the design methods that are described in Sect. 10.2 have been applied by this author to contrived or real-world applications, so this is left to future researchers. It is important, however, to understand that, just as there is a hierarchy of approaches to designing a singleton IT2 fuzzy system, there is also a hierarchy of approaches to designing a singleton[18] WH GT2 fuzzy system, namely:

[18]Because formulas have only been given in this chapter for singleton fuzzification, this is the only kind of fuzzification that is used in the discussions below. Exercises 11.24 and 11.25 ask the reader to extend these discussions to T1 non-singleton and IT2 non-singleton WH GT2 fuzzy systems.

Table 11.8 Example of choices (shaded) made to specify or design a WH GT2 TSK fuzzy system

Choices for	Kind of WH GT2 fuzzy system							
• Rules	WH-Mamdani				WH-TSK			
					A2-C0	A2-C1		
• Fuzzifier	Singleton	T1 non-singleton	IT2 non-singleton		Singleton	T1 non-singleton	IT2 non-singleton	
• *t*-norm	Minimum		Product		Minimum	Product		
• Combining fired rule output sets	Union		None		NA			
• Type-reduction +D	Centroid	Height	COS	NA	Unnormalized	Normalized		
• Direct D	WH-NT	WH-BMM	NA		NA			
• Number of input variables	p				3			
• Number and names of terms per input	$\{X_{1j}\}_{j=1}^{Q_1}$	$\{X_{2j}\}_{j=1}^{Q_2}$	…	$\{X_{pj}\}_{j=1}^{Q_p}$	*light*	*moderate*	*heavy*	
• Number and names of terms for, or structure of, output	$\{Y_j\}_{j=1}^{Q_y}$				$\{c_0^l + c_1^l x_1 + c_2^l x_2 + c_3^l x_3\}_{l=1}^{M}$			
• Number of rules	M				27			
• Kind of FOU[a]	Specify it		NA		Specify it	NA		
• Kind of UMFs	Triangle	Trapezoid	Other	NA	Triangle	Trapezoid	Other	NA
• Kind of LMFs	Triangle	Trapezoid	Other	NA	Triangle	Trapezoid	Other	NA
• FOU parameters	Pre-specified		Optimized		Pre-specified	Optimized		
• Kind of secondary MF	Triangle	Trapezoid	Other		Triangle	Trapezoid	Other	
• Secondary MF parameters	Pre-specified	Optimized	NA		Pre-specified	Optimized	NA	
• Squishing parameters	Pre-specified	Optimized	NA		Pre-specified	Optimized	NA	
• Number of horizontal slices	$k_{\max}$				5			

[a]If "Kind of FOU" is specified, then "Kind of UMFs" and "Kind of LMFs " must be specified as "NA." If "Kind of FOU" is "NA" then both "Kind of LMF" and "Kind of UMF" must be specified.

1. **Partially dependent approach**: In this approach, one first designs the best possible IT2 fuzzy system by tuning all of its parameters, and then uses these parameters in some way to *initialize* the parameters of the singleton WH GT2 fuzzy system.
2. **Totally independent approach**: In this approach, all of the parameters of the singleton WH GT2 fuzzy system are tuned without the benefit of a previous IT2 design. In this design, the parameters of the singleton WH GT2 fuzzy system are usually initialized in a random manner.

When a particle-based method is used to design a WH GT2 fuzzy system, the following three-step approach is advocated[19]:

1. Design a singleton T1 fuzzy system using a particle-based method (e.g., QPSO), and initialize all of its particles randomly. If system performance is acceptable, stop; otherwise proceed to Step 2.
2. Design a singleton IT2 fuzzy system by optimizing its parameters using a particle-based method in which one particle is associated with the just designed

[19]Because formulas have only been given in this chapter for singleton fuzzification, the non-singleton steps that are listed in Sect. 10.2.5 are not included here. Of course, once the formulas for T1 or IT2 non-singleton fuzzification have been obtained (Exercise 11.1) then more steps can be added to the present list, as has been done in Sect. 10.2.5.

T1 fuzzy system, and where the parameters of all the remaining particles are initialized randomly. If system performance is acceptable, stop; otherwise proceed to Step 3.

3. Design a singleton WH GT2 fuzzy system by optimizing its parameters using a particle-based method in which one particle is associated with the just designed IT2 fuzzy system, and where the parameters of all the remaining particles are initialized randomly. If system performance is acceptable, stop; otherwise choose a different kind of and try again.

Example 11.10 This example is a continuation of Example 10.10. To begin, one must set up the particle for a WH GT2 fuzzy system and then figure out how to embed an IT2 fuzzy system into such a particle. In order to set up the particle for a WH GT2 fuzzy system, many design choices have to be made. In order to preserve continuity in going from Example 10.10 to this example, our focus is on a (singleton) WH GT2 Mamdani fuzzy system that uses COS type-reduction + average of end-points defuzzification (Sect. 11.8.2), and the same design choices are made here as were made in Example 10.10, just for illustrative purposes. Those design choices are: all FOUs are for Gaussian primary MFs with uncertain standard deviations, so that each FOU is described by three design parameters $\{m, \sigma_1, \sigma_2\}$. In addition, the vertical-slice representation is used and a parsimonious choice is made for the secondary MFs, one that lets an IT2 FS be embedded in a GT2 FS.

Recall Example 6.22, which was for symmetrical trapezoid secondary MFs. Using the notation for antecedent GT2 FSs in (11.1), the vertices of the base of each trapezoid are located at $(i = 1, \ldots, M; j = 1, \ldots, p)$ $\underline{\mu}_{\tilde{F}_j^i}(x_j)$ and $\bar{\mu}_{\tilde{F}_j^i}(x_j)$, and its top is defined by left and right end-points, $EP_{j,l}^i(u|x_j)$ and $EP_{j,r}^i(u|x_j)$ [see Fig. 6.25, Eqs. (6.55) and (6.56)], that are parameterized as $(w_j^i \in [0, 1])$:

$$EP_{j,l}^i(u|x_j) = \underline{\mu}_{\tilde{F}_j^i}(x_j) + \frac{1}{2}w_j^i[\bar{\mu}_{\tilde{F}_j^i}(x_j) - \underline{\mu}_{\tilde{F}_j^i}(x_j)] \tag{11.89}$$

$$EP_{j,r}^i(u|x_j) = \bar{\mu}_{\tilde{F}_j^i}(x_j) - \frac{1}{2}w_j^i[\bar{\mu}_{\tilde{F}_j^i}(x_j) - \underline{\mu}_{\tilde{F}_j^i}(x_j)] \tag{11.90}$$

When $w_j^i = 0$, the trapezoid reduces to a square well, and the GT2 FS reduces to an IT2 FS, something that is crucial to the embedding of an IT2 fuzzy system into a WH GT2 fuzzy system particle. Hence, the choice for a secondary MF as a symmetrical trapezoid. The α-cuts for such T1 FSs are given in (6.57), from which it can be observed that once $\underline{\mu}_{\tilde{F}_j^i}(x_j)$, $\bar{\mu}_{\tilde{F}_j^i}(x_j)$ and w_j^i have been specified, then the end-points of the α-cuts, $a_{j,\alpha}^i$ and $b_{j,\alpha}^i$ can be computed by (6.56).

Consequently, the structure of a particle for a WH GT2 fuzzy system is:

$$\boldsymbol{\phi}_{\text{WH GT2}} = col\left(\underbrace{\overbrace{m_1^1, \sigma_{1,1}^1, \sigma_{1,2}^1, w_1^1}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^1, \sigma_{p,1}^1, \sigma_{p,2}^1, w_p^1}^{\text{Antecedent } p}, \overbrace{\left\{c_l(\tilde{G}_{\alpha_k}^1), c_r(\tilde{G}_{\alpha_k}^1)\right\}_{k=1}^{k_{\max}}}^{\text{Consequent}}}_{\text{Rule1}}; \right.$$
$$\left. \ldots; \underbrace{\overbrace{m_1^M, \sigma_{1,1}^M, \sigma_{1,2}^M, w_1^M}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^M, \sigma_{p,1}^M, \sigma_{p,2}^M, w_p^M}^{\text{Antecedent } p}, \overbrace{\left\{c_l(\tilde{G}_{\alpha_k}^M), c_r(\tilde{G}_{\alpha_k}^M)\right\}_{k=1}^{k_{\max}}}^{\text{Consequent}}}_{\text{Rule}M}\right) \tag{11.91}$$

Each rule has p antecedents that are described by four parameters and a consequent that is described by the two end-points of its centroid for each of the $k_{\max}$ horizontal slices. Note, also, that average of end-points defuzzification [see (11.51)] does not introduce any new design parameters once numerical values have been chosen for $\alpha_1, \ldots, \alpha_{k_{\max}}$.

The singleton IT2 fuzzy system particle, which must be of the same length as this WH GT2 fuzzy system particle for it to be embedded in such a particle, begins with (11.91) and expresses it as:

$$\boldsymbol{\phi}_{\text{IT2}}^S = \text{col}\left(\underbrace{\overbrace{m_1^1, \sigma_{1,1}^1, \sigma_{1,2}^1, 0}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^1, \sigma_{p,1}^1, \sigma_{p,2}^1, 0}^{\text{Antecedent } p}, \overbrace{\{c_l(\tilde{G}^1), c_r(\tilde{G}^1)\}_{k=1}^{k_{\max}}}^{\text{Consequent}}}_{\text{Rule 1}}; \cdots \right.$$
$$\left. \underbrace{\overbrace{m_1^M, \sigma_{1,1}^M, \sigma_{1,2}^M, 0}^{\text{Antecedent 1}}, \ldots, \overbrace{m_p^M, \sigma_{p,1}^M, \sigma_{p,2}^M, 0}^{\text{Antecedent } p}, \overbrace{\{c_l(\tilde{G}^M), c_r(\tilde{G}^M)\}_{k=1}^{k_{\max}}}^{\text{Consequent}}}_{\text{Rule } M}\right) \tag{11.92}$$

Observe, in (11.92), that: (1) All of the FOU parameters are taken from the particle-algorithm optimized singleton IT2 fuzzy system design; (2) By setting $w_j^i = 0$ in each of the antecedent sub-particles, IT2 FSs can be expressed as GT2 FSs; (3) By setting $\left\{c_l(\tilde{G}_{\alpha_k}^i), c_r(\tilde{G}_{\alpha_k}^i)\right\}_{k=1}^{k_{\max}} = \{c_l(\tilde{G}^i), c_r(\tilde{G}^i)\}_{k=1}^{k_{\max}}$ for $i = 1, \ldots, M$, all consequent GT2 FSs reduce to IT2 FSs; and, (4) In this way, it is straightforward to embed a singleton IT2 fuzzy system particle into a WH GT2 fuzzy system particle.

Interestingly, while it was very parsimonious to represent each consequent in an IT2 fuzzy system by using the end-points of its centroid, this parsimony is lost when the same idea is applied to each consequent of a GT2 FS. Observe, in (11.91), that it takes $2k_{\max}$ centroid end-points to represent each consequent GT2 FS, and $2k_{\max}$ could be a large number. So, for the first time, it is more parsimonious to represent each consequent by its FOU parameters and secondary parameters, exactly as is done for each antecedent GT2 FS. This only has to be done during the design of the GT2 fuzzy system. During the operational phase of the GT2 fuzzy system, $\left\{c_l(\tilde{G}^i_{\alpha_k}), c_r(\tilde{G}^i_{\alpha_k})\right\}_{k=1}^{k_{\max}} (i = 1, \ldots, M)$ would be computed and stored, since these quantities do not depend upon $\mathbf{x}'$.

Exercise 11.26 asks the reader to set up WH GT2 fuzzy system particles for other kinds of WH GT2 fuzzy systems.

11.15 Applications

As of the writing of this book (2017), very few papers have been published about real-world applications of WH GT2 fuzzy systems, or any other kind of GT2 fuzzy system. The possible reasons for this are:

1. WH-GT2 fuzzy systems are more complicated to understand and program than are IT2 fuzzy systems. Hopefully, this chapter will make them easier to understand and program.
2. One must compare the performance of a WH GT2 fuzzy system with the performances from IT2 and T1 fuzzy systems, and ideally with a non-fuzzy system, to see if significant improvement in performance has been obtained by using a WH GT2 fuzzy system. It takes a lot of work and time to do this.
3. Many choices have to be made for the structures of T1, IT2 and GT2 fuzzy systems, so there is a risk factor involved when specific choices are made. Additionally, the choices made for an IT2 fuzzy system should be commensurate with those made for a T1 fuzzy system, and those made for a WH GT2 fuzzy system should be commensurate with those made for an IT2 fuzzy system. Hopefully, Tables 4.1, 10.1 and 11.6 will make this easier to do.
4. An optimization method should be used that permits the embedding of a lower fuzzy system design (e.g., T1 and IT2) into a higher fuzzy system design (e.g., WH GT2), so that guaranteed performance improvement for the higher fuzzy system design is attained (such a method will also indicate if it cannot be attained). Generally speaking, this rules out using any of the parameters obtained in the lower fuzzy system for the higher fuzzy system, except, perhaps for initialization purposes.
5. When any source of randomness is present, either in the system (it may be random), or data (it may be corrupted by measurement noise) or the optimization algorithm (it may involve using one or more random numbers), then Monte Carlo simulations are needed. When more than one source of randomness is present, then massive Monte Carlo simulations are needed, and they have to be done for all of the fuzzy (T1, IT2, and WH GT2) and non-fuzzy systems (ideally, using the same realizations of random quantities for each design). Bootstrapping techniques can be used (when appropriate) to reduce the number of Monte Carlo realizations that are needed. Doing this correctly takes a lot of time.

6. Performance comparisons should be done using statistical tests (Derrac et al. 2011), when appropriate.
7. Providing enough background about the different kinds of fuzzy systems that are used in a journal article (so that their results can be reproduced by others) takes a lot of space, and some journals have very severe page limits that make it impossible to do this. Hopefully, this book will make it easier, by allowing authors to refer to equations in it.

Although none of the very small number of journal articles that have been published to date about GT fuzzy system applications fully meet the requirements of the majority of items 2–7, it is still useful to note that they are: Kumbasar and Hagras (2015), Gaxiola et al. (2015), Almaraashi et al. (2016), and Castillo et al. (2016a, b).

11.16 Case Study: WH GT2 Fuzzy Logic Control

Recall that Sect. 4.7 provided a short early history of fuzzy control, explained what a type-1 fuzzy logic controller (FLC) is, and provided: a short introduction to fuzzy PID control, the general structure of a fuzzy PID controller, conventional and fuzzy PID control design methods, and some simulation results comparing T1 fuzzy PID control with PID control. Recall, also, that Sect. 10.8 explained what an IT2 FLC is, and provided: a general structure of an IT2 fuzzy PID controller, and some simulation results comparing four kinds of IT2 FPID controllers with T1 FPID and conventional PID control.

This section explains what a GT2 FLC is, and provides some simulation results that compare a GT2 fuzzy PID controller with an IT2 FPID controller, as well as with T1 FPID and conventional PID controllers. The comparisons are not for the same system that was used in Sects. 4.7 and 10.8, because it was found (by Prof. Kumbasar) that a GT2 FLC did not significantly improve the performance of an IT2 FLC for that system.

11.16.1 What Is a GT2 FLC?

The GT2 fuzzy systems in Fig. 11.1 can be used as a GT2 FLC. The defuzzified output of such a GT2 FLC can be used as the command to an actuator in the control system.

As already noted in Sects. 4.7.2 and 10.8.1 some sources of uncertainties that face real-world control systems (including fuzzy logic control systems) are uncertainties about: inputs to the FLC, control outputs due to changes in the actuator, changes in operating conditions, disturbances acting upon the system, linguistic meanings of the antecedent terms that are used in the controller's rules, and expert's disagreements about the rule's consequents.

In a GT2 FLC all of these uncertainties are modeled by the FOUs and secondary MFs of the antecedents and/or consequents of the rules, as well as by the kind of fuzzifier.

Kumbasar and Hagras (2015) provide the following additional motivation for considering a GT2 FLC over an IT2 FLC:

> [Although a smoother control surface may be generated by the IT2 FPI controller] … the IT2-FPI transient state and disturbance rejection performance may degrade in comparison with its type-1 and conventional counterparts (Wu and Tan 2010; Wu 2012). Although a smooth control surface is probably a common objective in industrial practice, the problem is that the resulting disturbance response may be unacceptable (too slow) since disturbances occurring around the steady state might cause a smaller control output change (Skogestad 2006). This problem is usually solved by a trade-off between control performance and robustness (Aleantara et al. 2013).
>
> GT2 fuzzy PI controllers can [may] provide an acceptable trade-off between the robust control performance of the IT2-FPI [controller] and the acceptable transient and disturbance rejection performance of the type-1 PI controllers. Thus the GT2 fuzzy PI controller might be able to enhance both the transient state and disturbance rejection performances while preserving the robustness of the T2 fuzzy controller.

Just as T1 and IT2 FLCs are variable structure controllers, so is a GT2 FLC, and just as T1 and IT2 FLCs have two architectures, Mamdani and TSK, a GT2 FLC also has those two architectures.

11.16.2 System Description[20]

The system to be controlled is nonlinear, its control is subject to a time delay (Mudi and Pal 1999), and it is described by:

$$\frac{d^2y(t)}{dt^2} + \frac{dy(t)}{dt} + 0.25y^2(t) = u(t - L) \tag{11.93}$$

In this section, the transient state performance of this system is examined for a reference trajectory that begins at a value of zero, and jumps up to a value of unity, and after more time jumps back down to a value of zero (these jumps provide step responses), for four controllers: PID, T1 FPID, IT2 FPID, and WH GT2 FPID. In (11.93), the nominal value of the time delay L is taken to be 0.1 s, and all controllers are designed only for the reference variation of jumping from zero to unity. The reference variation of jumping from unity to zero is also examined to study the robustness of the controllers for a different operating point. Robustness is examined also for a change in L, by applying the controllers that are designed for the nominal system to one in which L changes to 0.3 s.

[20]This section, as well as Sects. 11.16.3 and 11.16.4 were prepared by Prof. Tufan Kumbasar.

11.16.3 Controller Designs

The design of the PID controller used the PID + filter structure given in (4.80) in which the coefficients of the PID structure (K_c, τ_I, τ_D and τ_f) were chosen to minimize the integral absolute error (IAE)[21] using GA.[22] The final optimal parameters are: $K_c = 0.51$, $\tau_I = 3.01$, $\tau_D = 0.63$ and $\tau_f = 0.38$.

The T1-FPID controller was designed as explained in Sect. 4.7.3.2. It used the nine T1 Zadeh rules that are given in Table 4.14. The two antecedent's and consequent's MFs of these rules are the ones that are given in Fig. 4.22. K_e was set equal to unity, and then K_d, K_0 and K_1 were chosen to also minimize the IAE. The final optimal parameters (obtained using GA) are: $K_d = 1.2$, $K_0 = 0.8$ and $K_1 = 0.5$.

The IT2-FPID controller was designed as explained in Sect. 10.8.2. It used the nine IT2 Zadeh rules that are given in Table 10.16. The two antecedent's FOUs of these rules are the ones that are given in Fig. 10.20. The consequents were fixed at the values that are given in Table 10.16. The parameters that were tuned are the heights of each of the LMFs (a, b, and c in Fig. 10.20). They were found for a Mamdani IT2 fuzzy system that used COS type-reduction + defuzzification, and were chosen so that this nonlinear system would be robust with respect to the different operating points mentioned above, which was also done by minimizing the IAE. The final parameters (obtained using GA) are: $a_E = c_E = 0.2$, $b_E = 0.9$, $a_{\Delta E} = c_{\Delta E} = 0.4$ and $b_{\Delta E} = 0.35$.

The WH GT2-FPID controller used COS type-reduction, average of end-points defuzzification, and the nine IT2 Zadeh rules that are given in Table 10.16, in which the IT2 FSs were replaced by GT2 FSs, but the consequents were still the values given in that table. In order to keep this example relatively simple, the GT2 FSs only used three horizontal slices at $\alpha_1 = 0$, $\alpha_2 = 0.5$ and $\alpha_3 = 1$ (recall though, that the $\alpha_1 = 0$ slice does not contribute anything to the final defuzzified output; however, it can help to define the other horizontal slices). The horizontal-slice α-cuts for $\tilde{N}, \tilde{Z}$ and $\tilde{P}$ are given in Table 11.9, and are for the special kinds of squished horizontal slices that are described in Exercise 6.20.[23] The FOUs of all antecedent terms look like the ones in Fig. 10.20. The secondary MFs for $\tilde{N}(e)$, $\tilde{P}(e), \tilde{Z}(\Delta e), \tilde{N}(\Delta e)$, and $\tilde{P}(\Delta e)$ look like the one in Fig. 11.11a, whereas the secondary MFs for $\tilde{Z}(e)$ look like the one in Fig. 11.11b. The squishing parameters (γ_j) were all chosen so that this nonlinear system would be robust with respect to the different operating points mentioned above, again by minimizing the IAE.

---

[21]The design methodology presented in the first three lines of (4.81) cannot be employed here because of the nonlinear nature of the system in (11.93).

[22]All GA designs were achieved by using the MATLAB® GA toolbox, and used 20 iterations and 20 populations-default MATLAB GA parameter settings. MATLAB is a registered trademark of The MathWorks, Inc.

[23]Note that the squishing parameters for all of these horizontal slices can be deduced from the entries into Table 11.9; each is the coefficient of the left-end of the stated interval.

Table 11.9 Antecedent secondary MF horizontal slices

j	α_j	E		
		$\tilde{N}_{\alpha_j}(e)$	$\tilde{Z}_{\alpha_j}(e)$	$\tilde{P}_{\alpha_j}(e)$
1	0	$[\underline{\mu}_{\tilde{N}_{\alpha_1}}(e), \bar{\mu}_{\tilde{N}_{\alpha_1}}(e)]$	$[\underline{\mu}_{\tilde{Z}_{\alpha_1}}(e), \bar{\mu}_{\tilde{Z}_{\alpha_1}}(e)]$	$[\underline{\mu}_{\tilde{P}_{\alpha_1}}(e), \bar{\mu}_{\tilde{P}_{\alpha_1}}(e)]$
2	0.5	$[0.2\bar{\mu}_{\tilde{N}_{\alpha_1}}(e), \bar{\mu}_{\tilde{N}_{\alpha_1}}(e)]$	$[0.6\bar{\mu}_{\tilde{Z}_{\alpha_1}}(e), \bar{\mu}_{\tilde{Z}_{\alpha_1}}(e)]$	$[0.2\bar{\mu}_{\tilde{P}_{\alpha_1}}(e), \bar{\mu}_{\tilde{P}_{\alpha_1}}(e)]$
3	1	$[0.5\bar{\mu}_{\tilde{N}_{\alpha_1}}(e), \bar{\mu}_{\tilde{N}_{\alpha_1}}(e)]$	$[\bar{\mu}_{\tilde{Z}_{\alpha_1}}(e), \bar{\mu}_{\tilde{Z}_{\alpha_1}}(e)]$	$[0.5\bar{\mu}_{\tilde{P}_{\alpha_1}}(e), \bar{\mu}_{\tilde{P}_{\alpha_1}}(e)]$
j	α_j	ΔE		
		$\tilde{N}_{\alpha_j}(\Delta e)$	$\tilde{Z}_{\alpha_j}(\Delta e)$	$\tilde{P}_{\alpha_j}(\Delta e)$
1	0	$[\underline{\mu}_{\tilde{N}_{\alpha_1}}(\Delta e), \bar{\mu}_{\tilde{N}_{\alpha_1}}(\Delta e)]$	$[\underline{\mu}_{\tilde{Z}_{\alpha_1}}(\Delta e), \bar{\mu}_{\tilde{Z}_{\alpha_1}}(\Delta e)]$	$[\underline{\mu}_{\tilde{P}_{\alpha_1}}(\Delta e), \bar{\mu}_{\tilde{P}_{\alpha_1}}(\Delta e)]$
2	0.5	$[0.8\bar{\mu}_{\tilde{N}_{\alpha_1}}(\Delta e), \bar{\mu}_{\tilde{N}_{\alpha_1}}(\Delta e)]$	$[0.2\bar{\mu}_{\tilde{Z}_{\alpha_1}}(\Delta e), \bar{\mu}_{\tilde{Z}_{\alpha_1}}(\Delta e)]$	$[0.8\bar{\mu}_{\tilde{P}_{\alpha_1}}(\Delta e), \bar{\mu}_{\tilde{P}_{\alpha_1}}(\Delta e)]$
3	1	$[0.85\bar{\mu}_{\tilde{N}_{\alpha 1}}(\Delta e), \bar{\mu}_{\tilde{N}_{\alpha_1}}(\Delta e)]$	$[0.5\bar{\mu}_{\tilde{Z}_{\alpha_1}}(\Delta e), \bar{\mu}_{\tilde{Z}_{\alpha_1}}(\Delta e)]$	$[0.85\bar{\mu}_{\tilde{P}_{\alpha_1}}(\Delta e), \bar{\mu}_{\tilde{P}_{\alpha_1}}(\Delta e)]$

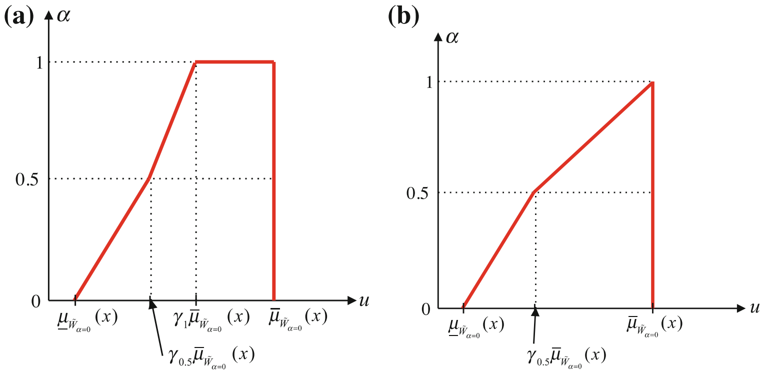


Fig. 11.11 Secondary MFs. **a** Used for $\tilde{N}(e)$, $\tilde{P}(e)$, $\tilde{Z}(\Delta e)$, $\tilde{N}(\Delta e)$, and $\tilde{P}(\Delta e)$, and **b** used for $\tilde{Z}(e)$

11.16.4 Simulation Results (WH GT2 FPID Versus IT2 FPID Versus T1 FPID and PID)

Responses for four situations are depicted in Fig. 11.12 for the *nominal system*. The left-hand step responses (going from zero to unity reference values) are for what all four PID controllers were designed for, whereas the right-hand step responses (going from unity to zero reference values) illustrate the robustness of the controlled nonlinear system with respect to a change in the operating point. % overshoot (OS), settling time (T_s) and IAE values are given in Table 11.10. As can be seen:

- *For the reference variation 0-1*: The WH GT2-FPID controller provides superior performance as compared to the conventional PID and T1 FPID controllers. Its percentage overshoot is significantly smaller than that from the IT2 FPID

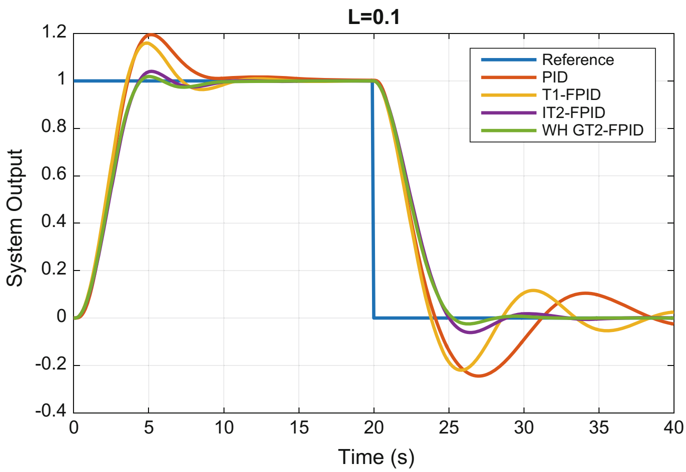


Fig. 11.12 System responses for the nominal system

Table 11.10 Performance measures for the nominal system

Controller	Reference variation 0–1			Reference variation 1–0		
	OS (%)	T_s(s)	IAE	OS (%)	T_s(s)	IAE
PID	19.6	8.6	29.4	24.4	>20	38.8
T1-FPID	15.9	9.7	26.2	22.0	>20	34.0
IT2-FPID	4.0	8.7	25.3	6.2	8.2	27.9
WH GT2-FPID	1.8	8.1	24.7	2.5	6.9	26.1

controller. Although its settling time and IAE are both smaller than those from the IT2 FPID controller, they are not "significantly" different.

- *For the reference variation 1-0*: The WH GT2 FPID controller provides vastly superior performance as compared to the conventional PID and T1 FPID controllers. Its percentage overshoot and settling time are vastly smaller than those from the conventional and T1 FPID controllers; and, its IAE is significantly smaller than that from the conventional, and T1 FPID controllers. Its percentage overshoot is significantly smaller than that from the IT2 FPID controller. The settling time and IAE from the WH GT2 FPID controller are smaller than those from the IT2 FPID controller, but the IAE may not be "significantly" different from the IAE of the IT2 FPID controller.

Responses for four situations are depicted in Fig. 11.13 for the *perturbed system*. Percentage overshoot, settling time, and integral absolute error values are given in Table 11.11. As can be seen:

Table 11.11 Performance measures for the perturbed system

Controller	Reference variation 0–1			Reference variation 1–0		
	OS (%)	T_s(s)	IAE	OS (%)	T_s(s)	IAE
PID	24.8	13.5	32.2	31.3	>20	45.6
T1-FPID	21.9	12.4	29.2	29.6	>20	41.8
IT2-FPID	8.9	9.0	26.7	10.4	11.3	29.6
WH GT2-FPID	7.2	8.4	25.8	6.7	9.7	27.2

- *For the reference variation 0–1*: The WH GT2-FPID controller provides superior performance as compared to the conventional PID and T1 FPID controllers. Its percentage overshoot, settling time and IAE are all smaller than those from the IT2-FPID controller, however, they are not "significantly" smaller.
- *For the reference variation 1–0*: The WH GT2-FPID controller continues to provide vastly superior performance as compared to the conventional PID and T1 FPID controllers. Its percentage overshoot, settling time, and IAE are all vastly smaller than those from the conventional and T1 FPID controllers. Its percentage overshoot is significantly smaller than that from the IT2 FPID controller. The settling time and IAE from the WH GT2 FPID controller are smaller than those from the IT2 FPID controller, but the IAE may not be "significantly" different from the IAE of the IT2 FPID controller.

In conclusion, although this is not a complete design of a WH GT2-FPID controller (such a design would examine more perturbed systems, the effects of using more than two active horizontal slices, and choosing other kinds of secondary MFs), the results demonstrate that improved control system performance is indeed possible when GT2 FSs are used as compared to using IT2 FSs.

Exercises

11.1 Explain how Theorem 11.1 changes for:

   (a) T1 non-singleton fuzzification
   (b) IT2 non-singleton fuzzification

11.2 Redo Examples 11.1 and 11.2 for a single-antecedent single-rule system: $\tilde{R}^1_Z$: IF x_1 is $\tilde{F}^1_1$ THEN y is $\tilde{G}^1$. Explain why $\tilde{B}_\alpha = \tilde{B}^1_\alpha$ and why COG type-reduction leads to the centroid of $\tilde{G}^1$, so that the output of this WH GT2 fuzzy system is independent of $x_1 = x'_1$. This is why Examples 11.1 and 11.2 used two antecedents and two rules.

11.3 Explain how the α-planes raised to level α were drawn in Fig. 11.5a, b.

11.4 Explain height type-reduction for a WH GT2 Mamdani fuzzy system in the same level of detail that centroid and COS type-reduction were described in Sects. 11.6.1 and 11.6.2, respectively, for such a fuzzy system.

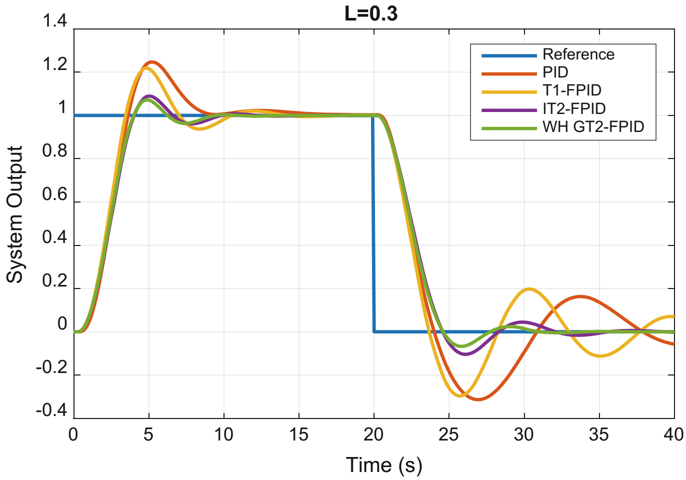


Fig. 11.13 System responses for the perturbed system

11.5 Explain type-reduction, in the same level of detail as in Sects. 11.6.3.1 and 11.6.3.2, for an:

(a) Unnormalized A2-C1 WH GT2 TSK fuzzy system
(b) Normalized A2-C1 WH GT2 TSK fuzzy system

11.6 In Example 11.3, explain why (11.44) is true.

11.7 Average end-points defuzzification formula (11.43) was stated for COS type-reduction. It was adapted for centroid type-reduction in (11.50), and for two kinds of GT2 TSK fuzzy systems in (11.52) and (11.53). State the adaptation of (11.43) for:

(a) Height type-reduction in a WH GT2 Mamdani fuzzy system
(b) Unnormalized A2-C1 WH GT2 TSK fuzzy system
(c) Normalized A2-C1 WH GT2 TSK fuzzy system

11.8 Continue the Sect. 11.9 comprehensive example by computing $y^{U}_{\text{WH-TSK}}(\mathbf{x}')$ and $y^{N}_{\text{WH-TSK}}(\mathbf{x}')$, using the consequents of the TSK rules that are in (9.137) and the numerical consequent coefficients that are given in the paragraph that is just below that equation (see, also, Table 9.3).

11.9 Repeat the Sect. 11.9 computations for $\mathbf{x}' = \text{col}(0.5, -0.5)$. Use triangle secondary MFs with an apex factor $w = 0.6$. Use the centroid α-cuts of the four consequent GT2 FSs that are given in Table 11.2. See, also Exercise 9.13.

11.10 Repeat the Sect. 11.9 computations for $\mathbf{x}' = \text{col}(-0.3, 0.6)$. Use triangle secondary MFs with an apex factor $w = 0.2$. Use the centroid α-cuts of the four consequent GT2 FSs that are given in Table 11.2.

11.11 Repeat the Sect. 11.9 computations for $\mathbf{x}' = \text{col}(0.5, -0.5)$. Use triangle secondary MFs with an apex factor $w = 0.2$. Use the centroid α-cuts of the four consequent GT2 FSs that are given in Table 11.2. See, also Exercise 9.13.

11.12 Extend the Sect. 9.8 WM UB Eqs. (9.142)–(9.152) to a WH GT2 WM UB fuzzy system.

11.13 Sect. 11.10.1 set up the computational steps for $y_{\text{NT},\alpha}(\mathbf{x}')$ in (11.64). Do the same for $y_{\text{NT},\alpha}(\mathbf{x}')$ in (11.62).

11.14 Section 11.10.1 set up the computational steps for $y_{\text{NT},\alpha}(\mathbf{x}')$ in (11.64). Do the same for $y_{\text{NT},\alpha}(\mathbf{x}')$ in (11.63).

11.15 Repeat the Section 11.11 computations for $\mathbf{x}' = \text{col}(0.5, -0.5)$. Use triangle secondary MFs with an apex factor $w = 0.2$. Use the centroid α-cuts of the four consequent GT2 FSs that are given in Table 11.2. Is $y_{\text{WH}-\text{NT}}(\mathbf{x}')$ different from the output of the NT IT2 fuzzy system that is given in (9.169)?

11.16 Continue the Sect. 11.11 comprehensive example by computing $y_{\text{BMM}}(\mathbf{x}')$, following the six-step procedure given in Sect. 11.10.2.

(a) What are the design choices you need to make?
(b) After making them, compute $y_{\text{BMM}}(\mathbf{x}')$, when $\mathbf{x}' = \text{col}(-0.3, 0.6)$. Because people will make different design choices, there can be multiple answers to this problem.

11.17 Explain why no GT2 FBF expansion exists for centroid type-reduction + defuzzification.

11.18 Obtain the GT2 FBF expansion and the GT2 FBFs for a WH GT2 Mamdani fuzzy system with height type-reduction + average of end-points defuzzification.

11.19 Obtain the GT2 FBF expansion and the GT2 FBFs for the WH A2-C0 normalized GT2 TSK fuzzy system.

11.20 Obtain the GT2 FBF expansion and the GT2 FBFs for the unnormalized WH A2-C1 GT2 TSK fuzzy system.

11.21 Obtain the GT2 FBF expansion and the GT2 FBFs for the normalized WH A2-C1 GT2 TSK fuzzy system.

11.22 Does a GT2 FBF expansion exist for a WH GT2 WM UB fuzzy system? If so, what is it?

11.23 How many additional layers, and what will their purpose be, in Fig. 11.10, for the following WH GT2 fuzzy systems that all use average of end-points defuzzification?

(a) Mamdani with centroid type-reduction
(b) Mamdani with height type-reduction
(c) Unnormalized A2-C0 TSK
(d) Normalized A2-C0 TSK
(e) Unnormalized A2-C1 TSK
(f) Normalized A2-C1 TSK

11.24 Explain the partially dependent and totally independent design approaches for *T1 non-singleton* WH GT2 fuzzy systems.

11.25 Explain the partially dependent and totally independent design approaches for *IT2 non-singleton* WH GT2 fuzzy systems.

11.26 Set up WH GT2 particles for the following WH GT2 fuzzy systems all of which use average of end-points defuzzification (exactly as was done in Example 11.10), and then explain how its comparable IT2 fuzzy system particle can be embedded in it:

(a) Mamdani with centroid type-reduction
(b) Mamdani with height type-reduction
(c) Unnormalized A2-C0 TSK
(d) Normalized A2-C0 TSK
(e) Unnormalized A2-C1 TSK
(f) Normalized A2-C1 TSK
(g) NT
(h) BMM

References

Almaraashi, M., R. John, A. Hopgood, and S. Ahmadi. 2016. Learning of interval and general type-2 fuzzy logic systems using simulated annealing: Theory and practice. *Information Sciences* 360: 21–42.

Aleantara, S., R. Vilanova, and C. Pedret. 2013. PID control in terms of robustness/performance and servo/regulator tradeoffs: A unifying approach to balanced autotuning. *Journal of Process Control* 23 (4): 527–542.

Castillo, O., L. A.-Angulo, J. R. Castro, and M. G.-Valdez. 2016. A comparative study of type-1 fuzzy logic systems, interval tyoe-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. *Information Sciences* 354: 257–274.

Castillo, O., L. Cervantes, J. Soria, M. Sanchez, and J.R. Castro. 2016b. A generalized type-2 fuzzy granular approach with applications to aerospace. *Information Sciences* 354: 165–177.

Derrac, J., S. Garcia, D. Molina, and F. Herrera. 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1: 3–18.

Gaxiola, F., P. Melin, F. Valdez, and O. Castillo. 2015. Generalized type-2 fuzzy weight adjustment for backpropagation neural networks in time series prediction. *Information Sciences* 325: 159–174.

Greenfield, S., and R. John. 2009. The uncertainty associated with a type-2 fuzzy set. In *Views on fuzzy sets and systems from different perspectives: Philosophy and logic, criticisms and applications*, ed. R. Seising, 471–483. Heidelberg: Springer.

Hamrawi, H., and S. Coupland. 2009. Type-2 fuzzy arithmetic using alpha-planes. In *Proceedings of the IFSA/EUSFLAT*, 606–611. Portugal.

Karnik, N.N., J.M. Mendel, and Q. Liang. 1999. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems* 7: 643–658.

Kumbasar, T., and H. Hagras. 2015. A self-tuning zslices based general type-2 fuzzy PI controller. *IEEE Transactions on Fuzzy Systems* 23: 991–1013.

Liu, F. 2008. An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. *Information Sciences* 178: 2224–2236.

Mendel, J.M. 2010. Type-2 fuzzy sets—A tribal parody. *IEEE Computational Intelligence Magazine* 5: 24–27.

Mendel, J.M. 2014. General type-2 fuzzy logic systems made simple: A tutorial. *IEEE Trans on Fuzzy Systems* 22: 1162–1182.

Mudi, R.K., and N.R. Pal. 1999. A robust self-tuning scheme for PI- and PD-type fuzzy controllers. *IEEE Transactions on Fuzzy Systems* 7 (1): 2–16.

Skogestad, S. 2006. Tuning for smooth PID control with acceptable disturbance rejection. *Industrial and Engineering Chemistry Research* 45 (23): 7817–7822.

Starczewski, J.T. 2009. Efficient triangular type-2 fuzzy logic systems. *International Journal of Approximate Reasoning* 50: 799–811.

Wagner, C., and H. Hagras. 2008. zSlices–Towards bridging the gap between interval and general type-2 fuzzy logic. In *Proceedings of the IEEE FUZZ conference*, Paper # FS0126. Hong Kong.

Wagner, C., and H. Hagras. 2010. Towards general type-2 fuzzy logic systems based on zSlices. *IEEE Transactions on Fuzzy Systems* 18: 637–660.

Wagner, C., and H. Hagras. 2013. ZSlices based general type-2 fuzzy sets and systems. In *Advances in type-2 fuzzy sets and systems: Theory and applications*, eds. Sadeghian, A., J. M. Mendel, and H. Tahayori. New York: Springer.

Wu, D. 2012. On the fundamental differences between interval type-2 and type-1 fuzzy logic controllers. *IEEE Transactions on Fuzzy Systems* 20: 832–848.

Wu, D., and W. W. Tan. 2010. Interval type-2 fuzzy PI controllers: Why they are more robust. In *Proceedings of IEEE international conference on granular computing*, 802–807. Silicon Valley.

Zhai, D., and J.M. Mendel. 2012. Comment on ‘toward general type-2 fuzzy logic systems based on zslices’. *IEEE Transactions on Fuzzy Systems* 20: 996–997.

Erratum to: Uncertain Rule-Based Fuzzy Systems

Erratum to:
J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
https://doi.org/10.1007/978-3-319-51370-6

In the original version of the book, extra server material has been included, extra's online logo has been added in the cover, and extra's link has been provided in the copyright page. The erratum book has been updated with the changes.

The updated online version of this book can be found at
https://doi.org/10.1007/978-3-319-51370-6

J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
https://doi.org/10.1007/978-3-319-51370-6_12

Index

J.M. Mendel, *Uncertain Rule-Based Fuzzy Systems*,
DOI 10.1007/978-3-319-51370-6