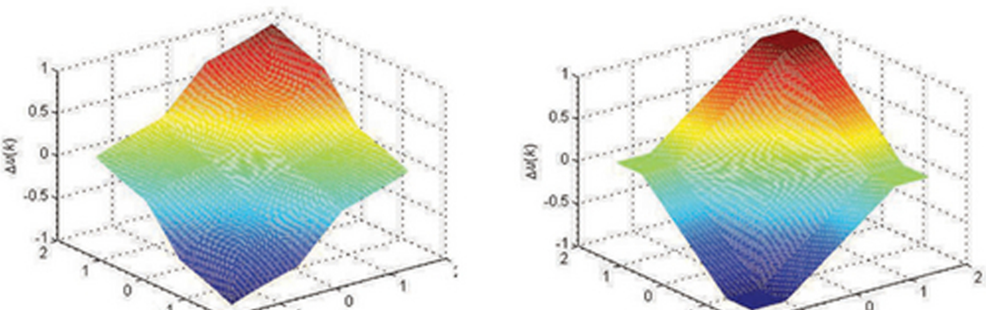
IEEE Press Series on Computational Intelligence
David B. Fogel, Series Editor

INTRODUCTION TO TYPE-2 FUZZY LOGIC CONTROL

THEORY AND APPLICATIONS



JERRY M. MENDEL, HANI HAGRAS, WOEI-WAN TAN,
WILLIAM W. MELEK, HAO YING

IEEE
IEEE PRESS

WILEY

INTRODUCTION TO TYPE-2 FUZZY LOGIC CONTROL

IEEE Press
445 Hoes Lane
Piscataway, NJ 08854

INTRODUCTION TO TYPE-2 FUZZY LOGIC CONTROL

THEORY AND APPLICATIONS

Jerry M. Mendel
Hani Hagras
Woei-Wan Tan
William W. Melek
Hao Ying

IEEE Press Series on Computational Intelligence
David B. Fogel, Series Editor

IEEE

WILEY

Published simultaneously in Canada

Library of Congress Cataloging-in-Publication Data:

Mendel, Jerry M., 1938–
Introduction to type-2 fuzzy logic control : theory and applications / Jerry M. Mendel, Hani Hagras, Woei-Wan Tan, William W. Melek, Hao Ying.
pages cm
Includes bibliographical references and index.
ISBN 978-1-118-27839-0 (cloth)
1. Automatic control. 2. Fuzzy systems. I. Hagras, Hani. II. Tan, Woei-Wan. III. Melek, William W. IV. Ying, Hao, 1958- V. Title.
TJ217.5.M46 2014
629.8′95633–dc23
2014000084

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To
the memory of
Ebrahim Mamdani (1943–2010)
Founder of Fuzzy Logic Control

CONTENTS

Preface xiii

Contributors xvii

1 Introduction 1

1.1 Early History of Fuzzy Control 1
1.2 What Is a Type-1 Fuzzy Set? 2
1.3 What Is a Type-1 Fuzzy Logic Controller? 3
1.4 What Is a Type-2 Fuzzy Set? 7
1.5 What Is a Type-2 Fuzzy Logic Controller? 9
1.6 Distinguishing an FLC from Other Nonlinear Controllers 10
1.7 T2 FLCs versus T1 FLCs 11
1.8 Real-World Applications of IT2 Mamdani FLCs 14
1.8.1 Applications to Industrial Control 14
1.8.2 Airplane Altitude Control 23
1.8.3 Control of Mobile Robots 24
1.8.4 Control of Ambient Intelligent Environments 27
1.9 Book Rationale 29
1.10 Software and How it Can Be Accessed 30
1.11 Coverage of the Other Chapters 30

2 Introduction to Type-2 Fuzzy Sets 32

2.1 Introduction 32
2.2 Brief Review of Type-1 Fuzzy Sets 32
2.2.1 Some Definitions 32
2.2.2 Set-Theoretic Operations 35
2.2.3 Alpha Cuts 36
2.2.4 Compositions of T1 FSs 39
2.2.5 Rules and Their MFs 40
2.3 Interval Type-2 Fuzzy Sets 42
2.3.1 Introduction 42

2.3.2 Definitions 43
2.3.3 Set-Theoretic Operations 51
2.3.4 Centroid of an IT2 FS 54
2.3.5 Properties of $c_l(k)$ and $c_r(k)$ 58
2.3.6 KM Algorithms as Well as Some Others 59
2.4 General Type-2 Fuzzy Sets 68
2.4.1 α-Plane/zSlice Representation 68
2.4.2 Set-Theoretic Operations 72
2.4.3 Centroid of a GT2 FS 73
2.5 Wrapup 77
2.6 Moving On 79

3 Interval Type-2 Fuzzy Logic Controllers 80

3.1 Introduction 80
3.2 Type-1 Fuzzy Logic Controllers 80
3.2.1 Introduction 80
3.2.2 T1 Mamdani FLCs 81
3.2.3 T1 TSK FLCs 85
3.2.4 Design of T1 FLCs 86
3.3 Interval Type-2 Fuzzy Logic Controllers 86
3.3.1 Introduction 86
3.3.2 IT2 Mamdani FLCs 87
3.3.3 IT2 TSK FLCs 103
3.3.4 Design of T2 FLCs 105
3.4 Wu–Mendel Uncertainty Bounds 105
3.5 Control Analyses of IT2 FLCs 111
3.6 Determining the FOU Parameters of IT2 FLCs 114
3.6.1 Blurring T1 MFs 114
3.6.2 Optimizing FOU Parameters 114
3.7 Moving On 122
Appendix 3A. Proof of Theorem 3.4 123
3A.1 Inner-Bound Set $[\overline{u}_l(\mathbf{x}), \underline{u}_r(\mathbf{x})]$ 123
3A.2 Outer-Bound Set $[\underline{u}_l(\mathbf{x}), \overline{u}_r(\mathbf{x})]$ 124

4 Analytical Structure of Various Interval Type-2 Fuzzy PI and PD Controllers 131

4.1 Introduction 131
4.2 PID, PI, and PD Controllers and Their Relationships 134

4.2.1 Two Forms of PID Controller—Position Form and Incremental Form 134
4.2.2 PI and PD Controllers and Their Relationship 135
4.3 Components of the Interval T2 Fuzzy PI and PD Controllers 136
4.4 Mamdani Fuzzy PI and PD Controllers—Configuration 1 140
4.4.1 Fuzzy PI Controller Configuration 140
4.4.2 Method for Deriving the Analytical Structure 144
4.5 Mamdani Fuzzy PI and PD Controllers—Configuration 2 154
4.6 Mamdani Fuzzy PI and PD Controllers—Configuration 3 162
4.6.1 Fuzzy PI Controller Configuration 162
4.6.2 Method for Deriving the Analytical Structure 165
4.7 Mamdani Fuzzy PI and PD Controllers—Configuration 4 169
4.7.1 Fuzzy PI Controller Configuration 169
4.7.2 Method for Deriving the Analytical Structure 171
4.8 TSK Fuzzy PI and PD Controllers—Configuration 5 181
4.8.1 Fuzzy PI Controller Configuration 181
4.8.2 Deriving the Analytical Structure 184
4.9 Analyzing the Derived Analytical Structures 185
4.9.1 Structural Connection with the Corresponding T1 Fuzzy PI Controller 186
4.9.2 Characteristics of the Variable Gains of the T2 Fuzzy PI Controller 190
4.10 Design Guidelines for the T2 Fuzzy PI and PD Controllers 194
4.10.1 Determination of θ_1 and θ_2 Values 196
4.10.2 Determination of the Remaining Nine Parameter Values 197
4.11 Summary 198
Appendix 4A 200

5 Analysis of Simplified Interval Type-2 Fuzzy PI and PD Controllers 205

5.1 Introduction 205
5.2 Simplified Type-2 FLCs: Design, Computation, and Performance 206
5.2.1 Structure of a Simplified IT2 FLC 207
5.2.2 Output Computation 208
5.2.3 Computational Cost 209
5.2.4 Genetic Tuning of FLC 210

5.2.5 Performance 211
5.2.6 Discussions 216
5.3 Analytical Structure of Interval T2 Fuzzy PD and PI Controller 221
5.3.1 Configuration of Interval T2 Fuzzy PD and PI Controller 221
5.3.2 Analysis of the Karnik–Mendel Type-Reduced IT2 Fuzzy PD Controller 227
5.3.3 Analysis of the IT2 Fuzzy PD Controller 231
5.4 Conclusions 248

6 On the Design of IT2 TSK FLCs **251**

6.1 Introduction 251
6.2 Preliminaries 251
6.2.1 Discrete T1 TSK FLC: Rules and Firing Level 252
6.2.2 Continuous T1 TSK FLC: Rules and Firing Level 252
6.2.3 T1 TSK FLC Output 253
6.2.4 Discrete IT2 TSK FLC: Rules and Firing Interval 253
6.2.5 Continuous IT2 TSK FLC: Rules and Firing Interval 253
6.2.6 IT2 TSK FLC Output 254
6.3 Novel Inference Engine for Control Design 254
6.4 Stability of IT2 TSK FLCs 255
6.4.1 Stability of Discrete IT2 TSK FLC 255
6.4.2 Stability of Continuous IT2 TSK FLC 258
6.4.3 Examples 259
6.5 Design of Adaptive IT2 TSK FLC 264
6.5.1 Rule Bases 264
6.5.2 Membership Functions 265
6.5.3 Control Structure 265
6.5.4 Control Design 266
6.5.5 Control Performance 267
6.6 Adaptive Control Design with Application to Robot Manipulators 268
6.6.1 Tracking Control 269
6.6.2 Control Structure 270
6.6.3 Application to Modular and Reconfigurable Robot Manipulators (MRR) 274

6.7 Robust Control Design 277
6.7.1 System Description 277
6.7.2 Disturbance Rejection Problem and Solution 280
6.7.3 Robust Control Example 284
6.8 Summary 285
Appendix 285

7 Looking into the Future 290

7.1 Introduction 290
7.2 William Melek and Hao Ying Look into the Future 290
7.3 Hani Hagras Looks into the Future 293
7.3.1 Nonsingleton IT2 FL Control 293
7.3.2 zSlices-Based Singleton General T2 FL Control 299
7.4 Woei Wan Tan Looks into the Future 306
7.5 Jerry Mendel Looks into The Future 307
7.5.1 IT2 FLC 307
7.5.2 GT2 FLC 309

Appendix A T2 FLC Software: From Type-1 to zSlices-Based General Type-2 FLCs 315

A.1 Introduction 315
A.2 FLC for Right-Edge Following 315
A.3 Type-1 FLC Software 316
A.3.1 Define and Set Up T1 FLC Inputs 316
A.3.2 Define T1 FSs That Quantify Each Variable 316
A.3.3 Define Logical Antecedents and Consequents for the FL Rules 318
A.3.4 Define Rule Base of T1 FLC 318
A.4 Interval T2 FLC Software 321
A.4.1 Define and Set Up FLC Inputs 323
A.4.2 Define IT2 FSs That Quantify Each Variable 323
A.4.3 Define Logical Antecedents and Consequents for the FL Rules 323
A.4.4 Define Rule Base of the IT2 FLC 323
A.5 zSlices-Based General Type-2 FLC Software 327
A.5.1 Define and Set Up FLC Inputs 327

A.5.2 Define zSlices-Based GT2 FSs That Quantify Each Variable 327
A.5.3 Define Logical Antecedents and Consequents for the FL Rules 335
A.5.4 Define Rule Base of the GT2 FLC 335

References **338**

Index **347**

PREFACE

When Lotfi Zadeh invented fuzzy sets in 1965, he never dreamt that the field in which they would be most widely used would arguably be the one that became the most hostile to the concept of fuzziness, namely control. Perhaps this was because the word “fuzzy” in Western civilization does not have a positive connotation and suggests an abandonment of mathematical rigor, one of the cornerstones of control. Perhaps it was because some famous mathematical probabilists (incorrectly) claimed that there was no difference between a fuzzy set and subjective probability. Perhaps it was because for almost a decade, until the 1974 seminal paper by Prof. Ebrahim Mamdani, who founded the field of fuzzy logic control and to whose memory our book is dedicated, there were no substantial real-world applications for fuzzy sets. Or, perhaps, it was because after the founding of this field many exaggerated claims were made by the fuzzy logic control community that flew in the face of mathematical rigor and did not pay attention to the same metrics that were and still are the cornerstones for control and cannot be ignored.

Now, 40 years after Mamdani’s seminal paper, fuzzy logic control using regular (i.e., type-1) fuzzy sets and logic has been extensively studied, applied to practical problems, and is very widely used in many real-world applications. It can and has been studied with the same level of mathematical rigor that control theorists are accustomed to, and is now considered a matured field; however, it still has some shortcomings. Its major shortcoming (in the opinions of the authors of this book) goes back to one of the earliest criticisms made about a type-1 fuzzy set, namely the unfuzziness of its membership function, that is, the word “fuzzy” has the connotation of being uncertain. But how can this connotation be captured by a membership function that is completely certain?

Importantly, in 1975 Zadeh introduced more general kinds of fuzzy sets in which their membership function grades are themselves fuzzy. The two most widely studied of these are *interval-valued fuzzy sets* and *type-2 fuzzy sets*. For the former, the membership grade is a uniformly weighted interval of values, whereas for the latter the membership grade is a nonuniformly weighted interval of values. Obviously, interval-valued fuzzy sets are a special case of type-2 fuzzy sets and are therefore called by many (as we do in this book) *interval type-2 fuzzy sets*.

Why should using type-2 fuzzy sets be of interest to the fuzzy logic control community? This question is answered in great detail in this book, but two short answers are: (1) they are more robust to system uncertainties and can provide better control system performance than type-1 fuzzy sets; and (2) there is now more than a critical mass of papers that have been published that demonstrate these improvements for many real-world applications.

Because of the lack of basic calculation methods for type-2 fuzzy sets in their early days, type-2 fuzzy logic controllers (T2 FLCs) did not emerge until fairly recently. Things have changed a lot during the past decade, so that type-2 fuzzy logic control (which is still an emerging field) now has the attention of the fuzzy systems community, and, as a result of this, the number of publications on it is growing quickly.

Recall that the central themes of any control methodology, fuzzy or conventional, are (1) to analyze various aspects of a control system and (2) to design a control system to achieve given user specifications. This book focuses on both topics for T2 FLCs and type-2 fuzzy logic control systems. The analysis includes (1) the mathematical structure of some T2 FLCs, (2) stability of type-2 fuzzy logic control systems, and (3) robustness of the type-2 fuzzy logic control systems.

This book, the first one entirely on T2 FLC, shows how to design type-2 fuzzy logic control systems based on a variety of choices for the T2 FLC components and also demonstrates how to apply type-2 fuzzy logic control theory to applications. It has been written by five of the leading experts on type-2 fuzzy sets, systems, and control, with the help of six contributors. It will be useful to any technical person interested in learning type-2 fuzzy logic control theory and its applications, from students to practicing engineers.

This is an introductory book that provides theoretical, practical, and application coverage of type-2 fuzzy logic control, and uses a coherent structure and uniform mathematical notations to link chapters, which are closely related, reflecting the book's central themes—analysis and design of type-2 fuzzy logic control systems. It has been written with an educational focus rather than a pure research focus. Each chapter includes worked examples, and most refer to their computer codes (programs) accessible through the book's common website, and outline how to use them at some high level. It is a self-contained reference book suitable for engineers, researchers, and college graduate students who want to gain deep insights about type-2 fuzzy logic control.

The book begins with an easy-to-read chapter meant to whet the reader's appetite so that he or she will read on; it explains what the differences are between a type-1 fuzzy set and a type-2 fuzzy set, and a T2 FLC and a T1 FLC, and, it provides many real-world applications in which T2 FLCs have shown marked improvements in performance over T1 FLCs. Chapter 2 provides all of the background material that is needed about type-2 fuzzy sets so that you can read the rest of the book; its main emphasis is on interval type-2 fuzzy sets because at present they are the most widely used type-2 fuzzy sets in type-2 fuzzy logic control. Chapter 3 is about Mamdani and TSK interval T2 FLCs. Chapter 4 examines the analytical structure of various interval type-2 fuzzy PI and PD controllers. Chapter 5 is about ways to simplify interval type-2 fuzzy PI and PD controllers. Chapter 6 is about the rigorous design of interval type-2 TSK fuzzy controllers. Chapter 7 provides each of the five authors with an opportunity to look into the future of type-2 fuzzy logic control. The book's appendix describes Java-based software that will let the reader examine type-1, interval type-2, and even general type-2 FLCs. All references (which are very extensive) have been integrated into one list that is at the end of the book.

The book's software can be downloaded by means of the following procedure: Software for Examples 4.1 and 4.6 and the examples in Chapter 6 can be accessed at http://booksupport.wiley.com, and software for Appendix A, that supports T1, IT2 and GT2 FLCs, is available at http://juzzy.wagnerweb.net.

In addition to the five authors, six of their (former) graduate students contributed to this book, to whom the authors are greatly appreciative. Their names are listed in the Contributors List. More specifically, Christian Wagner contributed to Chapters 2, 3 and 7, and prepared the entire Appendix; Xinyu Du and Haibo Zhou contributed to Chapter 4; Maowen Nie and Dongrui Wu contributed to Chapter 5; and Mohammad Biglarbegian contributed to Chapter 6.

The authors gratefully acknowledge material quoted from books or journals published by Elsevier, IEEE, John Wiley & Sons, Mancy Publishing (www.maney.co.uk/journals/irs and www.ingentaconnect.com/content/maney/ias) and Pearsons Education, Inc. For a complete listing of quoted books or articles, please see the References.

Jerry M. Mendel
Los Angeles, California

Hani Hagras
Colchester, UK

Woei-Wan Tan
Singapore

William W. Melek
Waterloo, Canada

Hao Ying
Detroit, Michigan

CONTRIBUTORS

Mohammad Biglarbegian, University of Guelph, Guelph Ontario, Canada

Xinyu Du, Wayne State University, Detroit, Michigan

Maowen Nie, A*Star Institute of Infocomm Research, Singapore

Christian Wagner, University of Nottingham, Nottingham, United Kingdom

Dongrui Wu, GE Global Research, New York

Haibo Zhou, Central South University, Changsha, China

CHAPTER 1

Introduction

1.1 EARLY HISTORY OF FUZZY CONTROL

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 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 proportional–integral–derivative (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.

Fuzzy logic controllers have been applied with great success to many real-world applications. The first FLC was developed by 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

Introduction to Type-2 Fuzzy Logic Control: Theory and Applications, First Edition.
Jerry M. Mendel, Hani Hagras, Woei-Wan Tan, William W. Melek, and Hao Ying.

Industry (MITI). The initiative established by MITI was a consortium called the 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 other applications. 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 United States, 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, microwaves, rice cookers, televisions, etc.), digital video cameras, cars, elevators (lifts), heavy industries (e.g., cement, petroleum, steel), and the like.

While this book focuses on type-2 fuzzy logic control, it will also provide background material about type-1 fuzzy logic control. Indeed, before we can explain what type-2 fuzzy logic control is we must briefly explain what type-1 fuzzy sets, type-1 fuzzy logic control, and type-2 fuzzy sets are. In this chapter we do this from a high-level perspective without touching on the mathematical aspects in order to give a feel for the nature of fuzzy sets and their applications. Later chapters in this book provide rigorous treatments of mathematical underpinnings of the subjects just mentioned.

1.2 WHAT IS A TYPE-1 FUZZY SET?

Suppose that a group of people is asked about the temperature values they associate with the linguistic concepts Hot and Cold. If *crisp sets* are employed, as shown in Fig. 1.1a, then a threshold must be chosen above which temperature values are considered Hot and below which they are considered Cold. Reaching a consensus about such a threshold is difficult, and even if an agreement can be reached—for example, 18°C—, is it reasonable to conclude that 17.99999°C is Cold whereas 18.00001°C is Hot?

On the other hand, Hot and Cold can be represented as *type-1 fuzzy sets* (T1 FSs) whose membership functions (MFs) are shown in Fig. 1.1b. Note that, prior to the appearance of type-2 fuzzy sets, the phrase *fuzzy set* was used instead of the phrase *T1 fuzzy set*. Even today, in many publications that focus only on T1 FSs, such sets are called fuzzy sets. In this book we shall use the phrase *type-1 fuzzy set*. Returning to Fig. 1.1b, observe that no sharp boundaries exist between the two sets

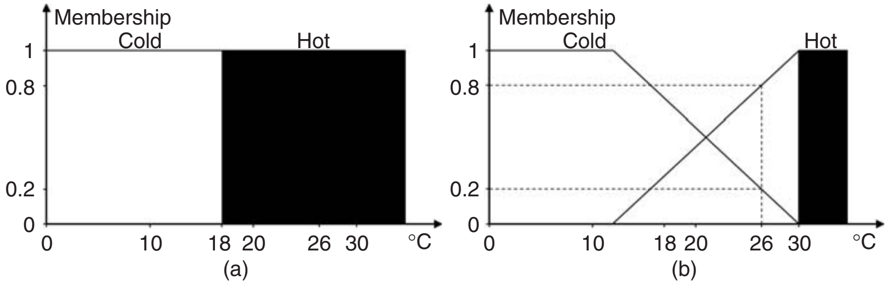


Figure 1.1 Representing Cold and Hot using (a) crisp sets, and (b) type-1 fuzzy sets.

and that each value on the horizontal axis may simultaneously belong to more than one T1 FS but with different degrees of membership. For example, 26°C, which is in the crisp Hot set with a membership value of 1.0 (Fig. 1.1a), is now in that set to degree 0.8, but is also in the Cold set to degree 0.2 (Fig. 1.1b).

Type-1 FSs provide a means for calculating intermediate values between the crisp values associated with being absolutely true (1) or absolutely false (0). Those values range between 0 and 1 (and can include them); thus, it can be said that a fuzzy set allows the calculation of shades of gray between white and black (or true and false). As will be seen in this book, the smooth transition that occurs between T1 FSs gives a good decision response for a type-1 fuzzy logic control system in the face of noise and other uncertainties.

1.3 WHAT IS A TYPE-1 FUZZY LOGIC CONTROLLER?

With the advent of type-2 fuzzy sets and type-2 fuzzy logic control, it has become necessary to distinguish between *type-2 fuzzy logic control* and all earlier fuzzy logic control that uses type-1 fuzzy sets (the distinctions between such fuzzy sets are explained in Section 1.4). We refer to fuzzy logic control that uses type-1 fuzzy sets as *type-1 fuzzy logic control*. When it does not matter whether the fuzzy sets are type-1 or type-2, we just use *fuzzy logic control* or *fuzzy control*.

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 degrees Celsius* 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 meters*, 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 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 its attempt to mimic human control actions, a type-1 FLC, whose structure is shown in Fig. 1.2, is composed of four main components: fuzzifier, rules, inference engine, and defuzzifier, where the operation of each component is summarized as follows:

- The fuzzifier maps each measured numerical input variable into a fuzzy set. One motivation for doing this is that measurements may be corrupted by noise and are somewhat uncertain (even after filtering). So, for example, a measured temperature of 26°C may be modeled as a triangular type-1 fuzzy set that is symmetrically centered around 26°C, where the base of the triangle is related to the uncertainty of this measurement. If, however, one believes that there is no measurement uncertainty, then the measurements can be modeled as crisp sets.
- Rules have an if–then structure, for example, *If Temperature is Low and Pressure is High*, *then Fan Speed is Low*. Each IF part of a rule is called its *antecedent*, and the THEN part of a rule is called its *consequent*. Rules relate input fuzzy sets to output fuzzy sets. All of the rules are collected into a rule base.

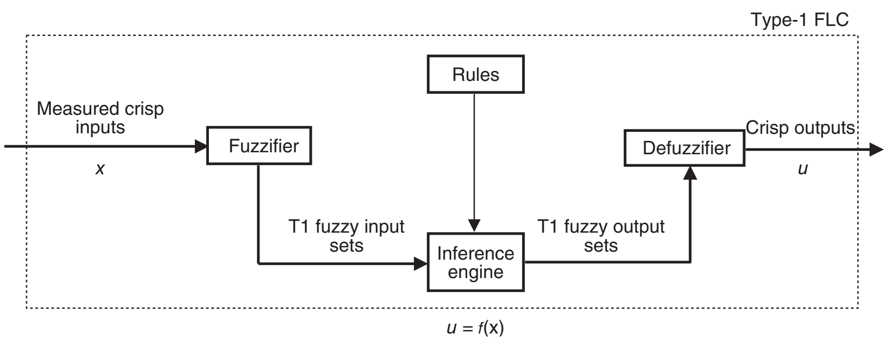


Figure 1.2 General structure of a type-1 FLC. The heavy lines with arrows indicate the path taken by signals during the actual operation of the FLC. Rules are used during the design of the FLC and are activated by the inference engine during the actual operation of the FLC (Mendel et al. (2006); © 2006, IEEE).

- The inference engine decides which rules from the rule base are fired and what their degrees of firing are, by using the fuzzy sets provided to it from the fuzzifier as well as some mathematics about fuzzy sets. The inference engine may also combine each rule's degree of firing with that rule's consequent fuzzy set to produce the rule's *output fuzzy set* (i.e., its fired-rule output set), and then combine all of those sets (across all of the fired rules) to produce an aggregated fuzzy output set using the mathematics of fuzzy sets; or it may send each rule's degree of firing directly to the defuzzifier where they are all aggregated in a different way.
- The defuzzifier receives either the aggregated fuzzy output sets from the inference engine or the degrees of firing for each rule plus some information about each consequent fuzzy set, and then processes this data to produce crisp outputs that are then passed to the physical actuators that control the actual plant.

In general, real-world control systems, such as 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, for example, wind, rain, humidity, and so forth. 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, and the like.
- 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, for example, wind buffeting an airplane.

In a T1 FLC all of these uncertainties are handled by the T1 FSs in the antecedents and consequents of the rules, as well as through the chosen type of fuzzifier. Regarding the latter, one may choose to use: (1) a singleton fuzzifier in which a measured value is treated as perfect and is modeled as a crisp set; or (2) a type-1 fuzzifier in which a measured value is treated as signal plus stationary noise and is modeled as a normal, convex T1 FS (also called a *T1 fuzzy number*).

The type-1 FLC in Fig. 1.2 is a nonlinear controller that maps its inputs $\mathbf{x}$ into an output u, that is, $u = f(\mathbf{x})$, 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 whose explicit mathematical expression is unknown in most cases.

Many researchers (e.g., Wang, 1992; Wang and Mendel, 1992a; Castro, 1995; Kosko, 1994; Kreinovich et al. 1998) have shown that the type-1 FLC $f(\mathbf{x})$ can uniformly approximate any real continuous function on a compact domain to any degree of accuracy; hence, FLCs are known to be *universal approximators*. One way to interpret what this means is that the FLC $f(\mathbf{x})$ approximates a function by covering its graph with *fuzzy patches* (Kosko, 1994), where each rule in the FLC defines a fuzzy patch in system's input–output space, and it then averages overlapping patches. This approximation improves as the fuzzy patches grow in number and shrink in size; however, as more smaller patches are included, the complexity of the model increases (i.e., the number of fuzzy sets and rules increases).

Type-1 FLCs produce nonlinear control laws $f(\mathbf{x})$ that cannot be effectively generated by any other mathematical means because such $f(\mathbf{x})$ are derived from linguistic if–then rules. 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* (VSC) is a form of discontinuous nonlinear control that alters the dynamics of a nonlinear system through the application of high-frequency switching control. A T1 FLC can also be regarded as a variable structure controller by virtue of the mathematics of fuzzy sets and systems; that is, 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 an FLC can be regarded as an extension of a conventional variable structure controller with a boundary layer.

There are two widely used architectures for a type-1 FLC that mainly differ in their fuzzy rule consequents. Those architectures, both of which are examined in this book, are:

- Mamdani FLC, developed by Mamdani and Assilian (1975) in which the antecedents and consequents of the rules are linguistic terms, for example: If x_1 *is Low and* x_2 *is High, then u is Low*. The linguistic labels in a Mamdani FLC are represented by type-1 fuzzy sets.
- Takagi–Sugeno (TS) FLC or Takagi–Sugeno–Kang (TSK) FLC (Takagi and Sugeno, 1985) in which the antecedents of the rules are also linguistic terms (modeled as type-1 fuzzy sets), but each rule's consequent is modeled as a mathematical function of the input variables, for example: *If* x_1 *is Low and* x_2 *is High, then* $u = g(x_1, x_2)$, where $g(x_1, x_2)$ is a polynomial function of x_1 and x_2 (this can include a constant, a linear or affine function, a quadratic function, etc.). An example of a first-order TSK FLC rule, the most widely used order, is: *If* x_1 *is Low and* x_2 *is High, then* $u = c_0 + c_1x_1 + c_2x_2$, where c_0, c_1, and c_2 are the consequent parameters.

1.4 WHAT IS A TYPE-2 FUZZY SET?

Because T1 FSs (e.g., as in Fig. 1.1b) are themselves crisp and precise (i.e., their MFs are supposedly known perfectly), this does not allow for any uncertainties about membership values, which is a potential shortcoming when using such fuzzy sets. A *type-2 fuzzy set* (T2 FS) is characterized by a fuzzy MF, that is, the membership value for each element of this set is itself a fuzzy set in [0,1]. The MFs of T2 FSs are three dimensional (3D) and include a *footprint of uncertainty* (FOU) (which is shaded in gray in Fig. 1.3a). It is the new third dimension of T2 FSs (e.g., Fig. 1.4c) and its FOU that provide additional degrees of freedom that make it possible to directly model and handle MF uncertainties.

In Fig. 1.3a, observe that the 26°C membership value in Hot is no longer a crisp value of 0.8 (as was the case in Fig. 1.1b); instead, it is a function that takes values from 0.6 to 0.8 in the primary membership domain, and maps them into a triangular distribution in the third dimension (Fig. 1.3b), called a *secondary MF*. This triangular secondary MF weights the interval [0.6, 0.8] more strongly over its middle values and less strongly away from those middle values. Of course, other weightings are possible, including equal weightings, in which case the T2 FS is called an *interval type-2 FS* (IT2 FS). Being able to choose different kinds of secondary MFs demonstrates one of the flexibilities of T2 FSs.

Figure 1.4c depicts the 3D MF of a general T2 FS whose secondary MFs [$f_x(u)$] are triangles. By convention, such a T2 FS is called a *triangular T2 FS*. Its FOU is depicted in Fig. 1.4a and its secondary MF at x' [$f_{x'}(u)$] is depicted by the solid triangle in Fig. 1.4b. When the secondary membership values equal 1 for all the primary membership values (as in the dashed curve in Fig. 1.4b), this results in an interval-valued secondary membership function, and, as just mentioned, the resulting T2 FS is called an IT2 FS. In Fig. 1.4c, $\mu(x, u)$ denotes the MF value at (x, u).

Figure 1.5 depicts the FOU of an IT2 FS for Low. The three dashed functions that are embedded within that FOU are T1 FSs. Clearly, one can cover this FOU with a multitude of such T1 FSs. At this point it is not important whether there are a

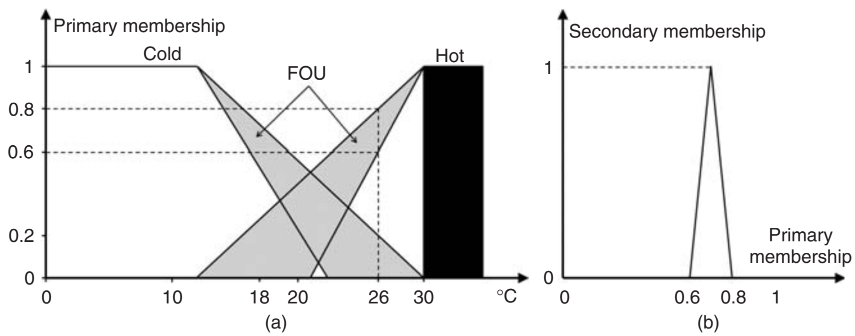


Figure 1.3 Type-2 fuzzy sets: (a) FOU and a primary membership and (b) a triangle secondary membership function.

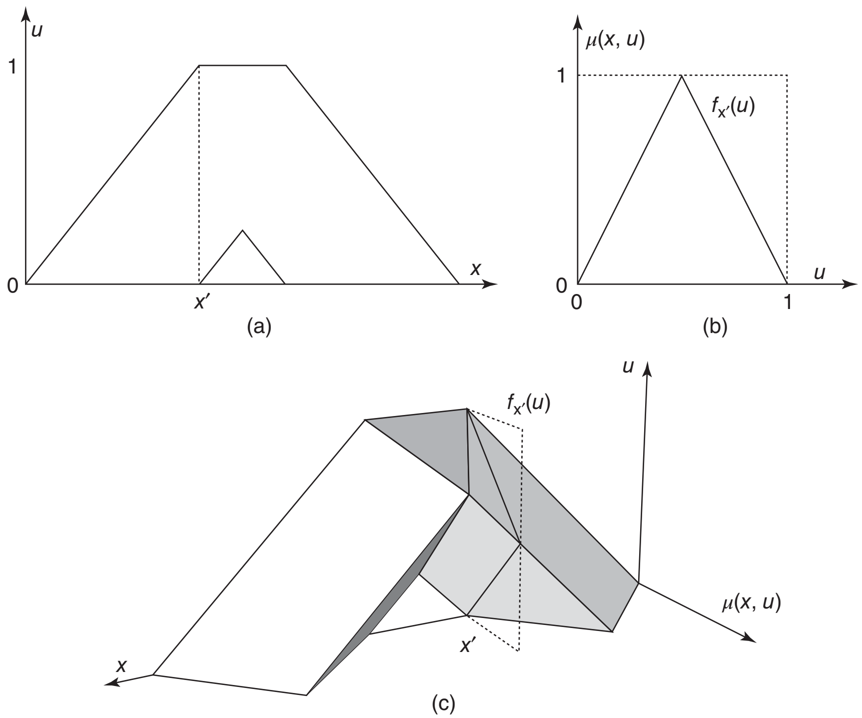


Figure 1.4 (a) FOU with primary membership (dashed) at x', (b) two possible secondary membership functions (triangle in solid line and interval in dashed line) associated with x', and, (c) the resulting 3D type-2 fuzzy set.

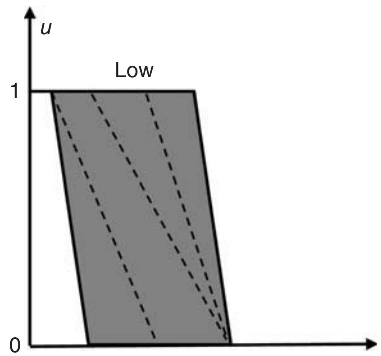


Figure 1.5 Three type-1 fuzzy sets that are embedded in the FOU of Low.

countable or uncountable number of such T1 FSs. What is important is interpreting an IT2 FS as the aggregation of a multitude of T1 FSs. This suggests that T1 FSs and everything that is already known about them can be used in derivations involving IT2 FSs, something that is exploited very heavily in this book. This interpretation also plays a very important role in understanding why an IT2 FLC may outperform a T1 FLC, something that we shall return to in the section below and in other chapters of this book.

1.5 WHAT IS A TYPE-2 FUZZY LOGIC CONTROLLER?

A type-2 FLC is depicted in Fig. 1.6. It contains five components: fuzzifier, rules, inference engine, type reducer, and defuzzifier. In a T2 FLC the inputs and/or outputs are represented by T2 FSs, and it operates as follows: crisp inputs, obtained from input sensors, are fuzzified into input T2 FSs, which then activate an inference engine that uses the same rules used in a T1 FLC to produce output T2 FSs. These are then processed by a type reducer that projects the T2 FSs into a T1 FS (this step is called *type reduction*) (Karnik et al., 1999; Liang and Mendel, 2000) after which that T1 FS is defuzzified to produce a crisp output that, for example, can be used as the command to an actuator in the control system. Type reduction followed by defuzzification is usually referred to as *output processing*.

In Section 1.3 we presented some sources of uncertainties that face real-world control systems in general. FLCs are also affected by:

- Linguistic uncertainties because the meaning of words that are used in the antecedents' and consequents' linguistic labels can be uncertain, that is, 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).

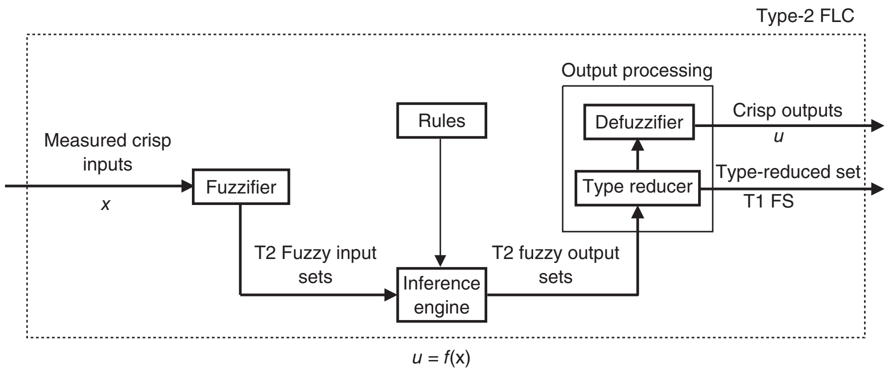


Figure 1.6 Overview of the architecture of a T2 FLC. The heavy lines with arrows indicate the path taken by signals during the actual operation of the FLC. Rules are used during the design of the FLC and are activated by the inference engine during the actual operation of the FLC (Mendel et al., 2006; © 2006, IEEE).

In a T2 FLC all of these uncertainties are modeled by the T2 FSs' MFs in the antecedents and/or consequents of the rules, as well as by the kind of fuzzifier. Regarding the latter, one may choose to use: (1) a singleton fuzzifier (as in a T1 FLC) in which a measured value is treated as perfect and is modeled as a crisp set; (2) a type-1 fuzzifier (as in a T1 FLC) in which a measured value is treated as signal plus stationary noise and is modeled as a normal, convex T1 FS (also called a *T1 fuzzy number*); or (3) a type-2 fuzzifier in which a measured value is treated as signal plus nonstationary noise and is modeled as a normal, convex T2 FS.

As we have explained in Section 1.4, a T2 FS can be thought of as a collection of many embedded T1 FSs (Mendel and John, 2002a). A T2 FLC may, therefore, be conceptually thought of as a collection of many (embedded) T1 FLCs whose crisp output is obtained by aggregating the outputs of all the embedded T1 FLCs (Karnik et al., 1999). Consequently, a T2 FLC has the potential to outperform a T1 FLC under certain conditions because it deals with uncertainties by aggregating a multitude of embedded T1 FLCs. The actual implementation of a T2 FLC does not actually require such an aggregation, but in this first chapter of this book, it is helpful to think of the output of a T2 FLC in this way.

Just as a T1 FLC is a variable structure controller so is a T2 FLC, and just as a T1 FLC has two architectures, Mamdani and TSK, a T2 FLC also has those two architectures. In a T2 Mamdani or TSK FLC, the fuzzy sets are type-2. Like their T1 FLC counterparts, T2 Mamdani and TSK FLCs are universal approximators (Ying, 2008, 2009). Both of these T2 FLC architectures will be covered in this book.

1.6 DISTINGUISHING AN FLC FROM OTHER NONLINEAR CONTROLLERS

Nonlinear control involves a nonlinear relationship between the controller's inputs and outputs and is more complicated than linear control; however, it is able to achieve better performance than linear control for many real-world control applications. Nonlinear control theory requires more challenging mathematical analysis and design than does linear control theory.

As mentioned in Section 1.3, an FLC is a nonlinear controller, that is, the function $f(\mathbf{x})$ is nonlinear. This will be demonstrated in later chapters of this book. What distinguishes an FLC, T1 or T2, from other nonlinear controllers is that it generates its nonlinear mapping function $f(\mathbf{x})$ through linguistic if–then rules and linguistic terms for the antecedents and consequents of the rules (e.g., Low Temperature, High Pressure). Such rules can be (easily) obtained from a human operator or can be postulated and learned from data. According to Kosko (1994), an FLC is unique in that it ties vague words like Low and High, and common sense rules, to state-space geometry.

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.