STUDIES IN *FUZZINESS*
AND *SOFT COMPUTING*

Studies in Fuzziness and Soft Computing

Oscar Castillo
Patricia Melin

Type-2 Fuzzy Logic: Theory and Applications

Springer

Oscar Castillo and Patricia Melin

Type-2 Fuzzy Logic: Theory and Applications

Studies in Fuzziness and Soft Computing, Volume 223

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 208. Oscar Castillo, Patricia Melin, Janusz Kacprzyk, Witold Pedrycz (Eds.)
Hybrid Intelligent Systems, 2007
ISBN 978-3-540-37419-0

Vol. 209. Alexander Mehler, Reinhard Köhler
Aspects of Automatic Text Analysis, 2007
ISBN 978-3-540-37520-3

Vol. 210. Mike Nachtegael, Dietrich Van der Weken, Etienne E. Kerre, Wilfried Philips (Eds.)
Soft Computing in Image Processing, 2007
ISBN 978-3-540-38232-4

Vol. 211. Alexander Gegov
Complexity Management in Fuzzy Systems, 2007
ISBN 978-3-540-38883-8

Vol. 212. Elisabeth Rakus-Andersson
Fuzzy and Rough Techniques in Medical Diagnosis and Medication, 2007
ISBN 978-3-540-49707-3

Vol. 213. Peter Lucas, José A. Gàmez, Antonio Salmerón (Eds.)
Advances in Probabilistic Graphical Models, 2007
ISBN 978-3-540-68994-2

Vol. 214. Irina Georgescu
Fuzzy Choice Functions, 2007
ISBN 978-3-540-68997-3

Vol. 215. Paul P. Wang, Da Ruan, Etienne E. Kerre (Eds.)
Fuzzy Logic, 2007
ISBN 978-3-540-71257-2

Vol. 216. Rudolf Seising
The Fuzzification of Systems, 2007
ISBN 978-3-540-71794-2

Vol. 217. Masoud Nikravesh, Janusz Kacprzyk, Lofti A. Zadeh (Eds.)
Forging New Frontiers: Fuzzy Pioneers I, 2007
ISBN 978-3-540-73181-8

Vol. 218. Masoud Nikravesh, Janusz Kacprzyk, Lofti A. Zadeh (Eds.)
Forging New Frontiers: Fuzzy Pioneers II, 2007
ISBN 978-3-540-73184-9

Vol. 219. Roland R. Yager, Liping Liu (Eds.)
Classic Works of the Dempster-Shafer Theory of Belief Functions, 2007
ISBN 978-3-540-25381-5

Vol. 220. Humberto Bustince, Francisco Herrera, Javier Montero (Eds.)
Fuzzy Sets and Their Extensions: Representation, Aggregation and Models, 2007
ISBN 978-3-540-73722-3

Vol. 221. Gleb Beliakov, Tomasa Calvo, Ana Pradera
Aggregation Functions: A Guide for Practitioners, 2007
ISBN 978-3-540-73720-9

Vol. 222. James J. Buckley, Leonard J. Jowers
Monte Carlo Methods in Fuzzy Optimization, 2008
ISBN 978-3-540-76289-8

Vol. 223. Oscar Castillo, Patricia Melin
Type-2 Fuzzy Logic: Theory and Applications, 2008
ISBN 978-3-540-76283-6

Oscar Castillo and Patricia Melin

Type-2 Fuzzy Logic: Theory and Applications

Springer

Authors

Prof. Dr. Oscar Castillo
Tijuana Institute of Technology
Division of Graduate Studies
Calzada Tecnologico S/N
Fracc. Tomas Aquino
Tijuana, B.C., 22379
Mexico
E-mail: ocastillo@tectijuana.mx

Prof. Dr. Patricia Melin
Tijuana Institute of Technology
Division of Graduate Studies
Calzada Tecnologico S/N
Fracc. Tomas Aquino
Tijuana, B.C., 22379
Mexico
E-mail: pmelin@tectijuana.mx

ISBN 978-3-540-76283-6 e-ISBN 978-3-540-76284-3

Studies in Fuzziness and Soft Computing ISSN 1434-9922

Library of Congress Control Number: 2007938042

© 2008 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting by the authors and Scientific Publishing Services Pvt. Ltd.

Printed in acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

We describe in this book, new methods for building intelligent systems using type-2 fuzzy logic and soft computing techniques. Soft Computing (SC) consists of several computing paradigms, including type-1 fuzzy logic, neural networks, and genetic algorithms, which can be used to create powerful hybrid intelligent systems. In this book, we are extending the use of fuzzy logic to a higher order, which is called type-2 fuzzy logic. Combining type-2 fuzzy logic with traditional SC techniques, we can build powerful hybrid intelligent systems that can use the advantages that each technique offers. We consider in this book the use of type-2 fuzzy logic and traditional SC techniques to solve pattern recognition problems in real-world applications. We consider in particular the problems of face, fingerprint and voice recognition. We also consider the problem of recognizing a person by integrating the information given by the face, fingerprint and voice of the person. Other types of applications solved with type-2 fuzzy logic and SC techniques, include intelligent control, intelligent manufacturing, and adaptive noise cancellation.

This book is intended to be a major reference for scientists and engineers interested in applying type-2 fuzzy logic for solving problems in pattern recognition, intelligent control, intelligent manufacturing, robotics and automation. This book can also be used as a textbook or major reference for graduate courses like the following: soft computing, intelligent pattern recognition, computer vision, applied artificial intelligence, and similar ones. We consider that this book can also be used to get novel ideas for new lines of research, or to continue the lines of research proposed by the authors of the book.

In Chapter 1, we begin by offering a brief introduction of the potential use of type-2 fuzzy logic in different real-world applications. We discuss the application of type-2 fuzzy logic in problems of pattern recognition. We also describe the use of type-2 fuzzy logic in problems of intelligent control of non-linear plants. We also outline the application of type-2 fuzzy logic in real-world applications of intelligent manufacturing, robotics and automation.

We describe in Chapter 2 the main ideas underlying type-1 fuzzy logic, and the application of this powerful computational theory to the problems of modeling, control and pattern recognition. We discuss in some detail type-1 fuzzy set theory, fuzzy reasoning, and fuzzy inference systems. At the end, we also give some general guidelines for the process of fuzzy modeling. We illustrate these concepts with several examples that show the applicability of type-1 fuzzy logic. The importance of type-1 fuzzy logic as a basis for developing intelligent systems has been recognized in several areas of application.

We describe in Chapter 3 the basic concepts, notation, and theory of type-2 fuzzy logic, which is a generalization of type-1 fuzzy logic. Type-2 fuzzy logic enables the management of uncertainty in a more complete way. This is due to the fact that in type-2 membership functions we also consider that there is uncertainty in the form of the functions, unlike type-1 membership functions in which the functions are considered to be fixed and not uncertain. We describe type-2 fuzzy set theory, type-2 fuzzy reasoning, and type-2 fuzzy systems. We also give examples to illustrate these ideas to the reader of the book.

We describe in Chapter 4 an efficient method for fuzzy inference in interval type-2 fuzzy logic. The new method of inference has been proposed to obtain fast output results of interval type-2 fuzzy systems. Fast and approximate results are justified in problems that need to work in real time. For example, fuzzy controllers that need to give solutions in real-time need obtain the outputs very fast. In this case, the new inference method uses only two values to approximate the interval fuzzy system. Details of this method are given in this chapter.

We describe in Chapter 5 the basic concepts, notation and learning algorithms for designing intelligent systems with interval type-2 fuzzy logic. Detailed methods for design and implementation of type-2 fuzzy systems are presented. The methods are illustrated with simple examples and are the basis for the applications described later. For this reason, we consider this chapter very important for understanding some of the applications that are described in later chapters of the book.

We describe in Chapter 6 a new approach for human recognition using as information the face, fingerprint, and voice of a person. Intelligent techniques can be used for achieving face recognition, fingerprint recognition, and voice identification. In this chapter we are considering the integration of these three biometric measures to improve the accuracy of human recognition. The new approach will integrate the information from three main modules, one for each of the three biometric measures. The new approach consists in a modular architecture that contains three basic modules: face, fingerprint, and voice. The final decision is based on the results of the three modules and uses interval type-2 fuzzy logic to take into account the uncertainty of the outputs of the modules.

We describe in Chapter 7 a new method for improving training in modular neural networks using interval type-2 fuzzy logic. In this chapter we consider two parts of a Modular Neural Network for image recognition, where a Type-2 Fuzzy Inference System (FIS 2) makes a great difference. The first FIS 2 is used for feature extraction in training data, and the second one to find the ideal parameters for the integration method of the modular neural network.

We describe in Chapter 8 a new method for edge border detection using interval type-2 fuzzy logic. Edges detection in digital images is a problem that has been solved by means of the application of different techniques from digital signal processing, also the combination of some of these techniques with Fuzzy Inference System (FIS) has been proposed. In this chapter a new Type-2 Fuzzy Logic method is implemented for the detection of edges and the results of three different techniques for the same task are compared.

We describe in Chapter 9 a method for the systematic design of stable type-2 fuzzy logic controllers. Stability is one of the more important aspects in the traditional knowledge of Automatic Control. Type-2 Fuzzy Logic is an emerging and promising area for achieving Intelligent Control (in this case, Fuzzy Control). In this chapter, we use the Fuzzy Lyapunov Synthesis as proposed by Margaliot to build a Lyapunov Stable Type-1 Fuzzy Logic Control System. Then we make an extension from a Type-1 to a Type-2 Fuzzy Logic Control System, ensuring the stability on the control system and proving the robustness of the corresponding fuzzy controller.

We describe in Chapter 10 an exhaustive experimental study of type-2 fuzzy logic controllers for different non-linear plants. Uncertainty is an inherent part in controllers used for real-world applications. The use of new methods for handling incomplete information is of fundamental importance in engineering applications. This chapter deals with the design of controllers using type-2 fuzzy logic for minimizing the effects of uncertainty produced by the instrumentation elements. We simulated type-1 and type-2 fuzzy logic controllers to perform a comparative analysis of the systems' response, in the presence of uncertainty.

We describe in Chapter 11 the evolutionary optimization of interval type-2 fuzzy controllers. The use of new methods for handling incomplete information is of fundamental importance in engineering applications. We simulated the effects of uncertainty produced by the instrumentation elements in type-1 and type-2 fuzzy logic controllers to perform a comparative analysis of the systems' response, in the presence of uncertainty. We are presenting an innovative idea to optimize interval type-2 membership functions using an average of two type-1 systems with the Human Evolutionary Model, we are showing comparative results of the optimized proposed method. We found that the optimized membership functions for the inputs of a type-2 system tend to increase the performance of the system for high noise levels.

We describe in Chapter 12 the design and implementation of interval type-2 fuzzy logic systems. This chapter presents the development and design of a graphical user interface and a command line programming Toolbox for construction, edition and simulation of Interval Type-2 Fuzzy Inference Systems. The Interval Type-2 Fuzzy Logic System Toolbox (IT2FLS) is an environment for interval type-2 fuzzy logic inference system development. Tools that cover the different phases of the fuzzy system design process, from the initial description phase, to the final implementation phase, constitute the Toolbox. The Toolbox's best qualities are the capacity to develop complex systems and the flexibility that allows the user to extend the availability of functions for working with the use of type-2 fuzzy operators, linguistic variables, interval type-2 membership functions, defuzzification methods and the evaluation of Interval Type-2 Fuzzy Inference Systems.

We describe in Chapter 13 the intelligent control of the pendubot using type-2 fuzzy logic and neural networks. First, the general concept of adaptive model-based control is described. Second, the use of type-2 fuzzy logic for adaptive control is described. Third, a neuro-fuzzy approach is proposed to learn the parameters of the fuzzy system for control. A specific non-linear plant was used to simulate the hybrid approach for adaptive control. The specific plant was also used as test bed in the experiments. The non-linear plant that was considered is the "Pendubot", which is a non-linear plant similar to the two-link robot arm. The results of the type-2 fuzzy logic approach for control were good, both in accuracy and efficiency.

We describe in Chapter 14 a method for automated quality control in sound speaker manufacturing using type-2 fuzzy logic and fractal theory. Traditional quality control has been done by manually checking the quality of sound after production. This manual checking of the speakers is time consuming and occasionally was the cause of error in quality evaluation. For this reason, we developed an intelligent system for automated quality control in sound speaker manufacturing. The intelligent system has a type-2 fuzzy rule base containing the knowledge of human experts in quality control. The parameters of the fuzzy system are tuned by applying neural networks using, as training data, a real time series of measured sounds as given by good sound speakers. We also use the fractal dimension as a measure of the complexity of the sound signal.

We describe in Chapter 15 a new approach for plant monitoring and diagnostics using type-2 fuzzy logic and fractal theory. The concept of the fractal dimension is used to measure the complexity of the time series of relevant variables for the process. A set of type-2 fuzzy rules is used to represent the knowledge for monitoring the process. In the type-2 fuzzy rules, the fractal dimension is used as a linguistic variable to help in recognizing specific patterns in the measured data. The fuzzy-fractal approach has been applied before in problems of financial time series prediction and for other types of problems, but now it is proposed to the monitoring of plants using type-2 fuzzy logic. We also compare the results of the type-2 fuzzy logic approach with the results of using only a traditional type-1 fuzzy logic approach. Experimental results show a significant improvement in the monitoring ability with the type-2 fuzzy logic approach.

We describe in Chapter 16 a new method for intelligent tracking of autonomous robotic systems using interval type-2 fuzzy logic and genetic algorithms. We develop a tracking controller for the dynamic model of unicycle mobile robot by integrating a kinematic controller and a torque controller based on Fuzzy Logic Theory. Computer simulations are presented confirming the performance of the tracking controller and its application to different navigation problems.

We describe in Chapter 17 the application of type-2 fuzzy logic for achieving adaptive noise cancellation. The objective of adaptive noise cancellation is to filter out an interference component by identifying a model between a measurable noise source and the corresponding un-measurable interference. In this chapter, we propose the use of type-2 fuzzy logic to find this model. The use of type-2 fuzzy logic is justified due to the high level of uncertainty of the process, which makes difficult to find appropriate parameter values for the membership functions.

We end this preface of the book by giving thanks to all the people who have help or encourage us during the writing of this book. First of all, we would like to thank our colleague and friend Prof. Janusz Kacprzyk for always supporting our work, and for motivating us to write our research work. We would also like to thank our colleagues working in Soft Computing, which are too many to mention each by their name. Of course, we need to thank our supporting agencies, CONACYT and DGEST, in our country for their help during this project. We have to thank our institution, Tijuana Institute of Technology, for always supporting our projects. Finally, we thank our families for their continuous support during the time that we spend in this project.

July 2007

Oscar Castillo
Patricia Melin
Tijuana, Mexico

Contents

1 Introduction to Type-2 Fuzzy Logic 1

2 Type-1 Fuzzy Logic 5

2.1 Type-1 Fuzzy Set Theory 6
2.2 Fuzzy Rules and Fuzzy Reasoning 11
2.2.1 Fuzzy Relations 11
2.2.2 Fuzzy Rules 13
2.3 Fuzzy Inference Systems 16
2.4 Fuzzy Modeling 27
2.5 Summary 28

3 Type-2 Fuzzy Logic 29

3.1 Type-2 Fuzzy Sets 29
3.2 Operations of Type-2 Fuzzy Sets 34
3.3 Type-2 Fuzzy Systems 37
3.3.1 Singleton Type-2 Fuzzy Logic Systems 38
3.3.2 Non-singleton Fuzzy Logic Systems 41
3.3.3 Sugeno Type-2 Fuzzy Systems 42
3.4 Summary 43

4 A Method for Type-2 Fuzzy Inference in Control Applications 45

4.1 Introduction 45
4.2 Proposed Method to Implement Type-2 FLC 46
4.3 Experiments 49
4.4 Results 49
4.5 Summary 52

5 Design of Intelligent Systems with Interval Type-2 Fuzzy Logic 53

5.1 Introduction 53
5.2 Fuzzy Logic Systems 54
5.2.1 Type-1 Fuzzy Logic Systems 54
5.2.2 Type-2 Fuzzy Logic Systems 56

5.2.2.1 Fuzzifier…………………………………………………… 59
5.2.2.2 Rules………………………………………………………. 59
5.2.2.3 Inference…………………………………………………. 59
5.2.2.4 Type Reducer……………………………………………. 60
5.2.2.5 Defuzzifier………………………………………………. 61
5.2.3 Performance Criteria…………………………………………… 61
5.3 Experimental Results for Intelligent Control………………………….. 62
5.4 Summary………………………………………………………………. 75

6 Method for Response Integration in Modular Neural Networks with Type-2 Fuzzy Logic……………………………………………………….. 77
6.1 Introduction…………………………………………………………... 77
6.2 Proposed Approach for Recognition………………………………….. 79
6.3 Modular Neural Networks…………………………………………….. 79
6.4 Integration of Results for Person Recognition Using Fuzzy Logic….. 80
6.5 Modular Neural Networks with Type-2 Fuzzy Logic as a Method for Response Integration…………………………………………………… 81
6.6 Simulation Results…………………………………………………… 82
6.7 Summary……………………………………………………………… 86

7 Type-2 Fuzzy Logic for Improving Training Data and Response Integration in Modular Neural Networks for Image Recognition……….. 87
7.1 Method for Image Recognition……………………………………….. 87
7.2 Type-2 Fuzzy Inference System as Edge Detector…………………….. 88
7.3 The Modular Structure………………………………………………… 91
7.4 Simulation Results……………………………………………………. 91
7.5 Summary……………………………………………………………… 94

8 Fuzzy Inference Systems Type-1 and Type-2 for Digital Images Edge Detection…………………………………………………………………. 95
8.1 Introduction…………………………………………………………… 95
8.2 Sobel Operators……………………………………………………….. 96
8.3 Edge Detection by Gradient Magnitude………………………………. 97
8.4 Edge Detection Using Type-1 Fuzzy Logic…………………………… 98
8.5 Edge Detection Using Type-2 Fuzzy Logic…………………………… 102
8.6 Comparison of Results………………………………………………… 103
8.7 Summary……………………………………………………………….. 107

9 Systematic Design of a Stable Type-2 Fuzzy Logic Controller………….. 109
9.1 Introduction…………………………………………………………… 109
9.2 Fuzzy Logic Controllers………………………………………………. 110
9.2.1 Type-1 Fuzzy Logic Control………………………………….... 110
9.2.2 Type-2 Fuzzy Logic Control…………………………………… 111
9.3 Systematic and Design of Stable Fuzzy Controllers………………….. 112

9.4 Experimental Results.. 116
9.5 Summary.. 120

10 Experimental Study of Intelligent Controllers Under Uncertainty Using Type-1 and Type-2 Fuzzy Logic.. 121

10.1 Introduction.. 121
10.2 Fuzzy Controllers.. 122
10.3 Experimental Results... 125
10.4 Summary.. 131

11 Evolutionary Optimization of Interval Type-2 Membership Functions Using the Human Evolutionary Model...................................... 133

11.1 Introduction.. 133
11.2 Fuzzy Controllers.. 134
11.3 The Human Evolutionary Model...................................... 135
11.4 Experimental Results... 138
11.5 Summary.. 144

12 Design of Fuzzy Inference Systems with the Interval Type-2 Fuzzy Logic Toolbox.. 145

12.1 Introduction.. 145
12.2 Interval Type-2 Fuzzy Set Theory.................................... 146
12.3 Interval Type-2 Fuzzy System Design............................... 149
12.4 Simulation Results... 150
12.5 Summary.. 153

13 Intelligent Control of the Pendubot with Interval Type-2 Fuzzy Logic... 155

13.1 Introduction.. 155
13.2 Adaptive Fuzzy Control.. 156
13.3 Type-2 Fuzzy Logic.. 159
13.4 Description of the Pendubot Plant.................................... 161
13.5 Simulation Results with Type-1 Fuzzy Control.................. 165
13.6 Type-2 Fuzzy Control of the Pendubot.............................. 167
13.7 Summary.. 170

14 Automated Quality Control in Sound Speakers Manufacturing Using a Hybrid Neuro-fuzzy-Fractal Approach.. 171

14.1 Introduction.. 171
14.2 Basic Concepts of Sound Speakers.................................. 172
14.2.1 Sound Basics... 172
14.2.2 Making Sound.. 173
14.2.3 Chunks of the Frequency Range.................................. 175
14.3 Type-2 Fuzzy Logic Systems... 175
14.3.1 Type-2 Fuzzy Sets.. 176
14.3.2 Type-2 Fuzzy Systems.. 177
14.4 Description of the Problem... 179

14.5 Fractal Dimension of an Object... 180
14.6 Experimental Results.. 181
14.7 Summary.. 185

15 A New Approach for Plant Monitoring Using Type-2 Fuzzy Logic and Fractal Theory.. 187

15.1 Introduction... 187
15.2 Monitoring and Diagnosis.. 188
15.3 Fractal Dimension of a Geometrical Object................................ 190
15.4 Fuzzy Estimation of the Fractal Dimension................................ 192
15.5 Type-2 Fuzzy Logic.. 192
15.6 Plant Monitoring Using a Type-1 Fuzzy-Fractal Approach............ 195
15.7 Plant Monitoring Using the Type-2 Fuzzy-Fractal Approach.......... 198
15.8 Summary.. 202

16 Intelligent Control of Autonomous Robotic Systems Using Interval Type-2 Fuzzy Logic and Genetic Algorithms.................................. 203

16.1 Introduction... 203
16.2 Problem Formulation... 204
16.3 Control of the Kinematic Model... 205
16.4 Fuzzy Logic Controller.. 206
16.5 Simulation Results.. 208
16.6 Summary.. 212

17 Adaptive Noise Cancellation Using Type-2 Fuzzy Logic and Neural Networks.. 213

17.1 Introduction... 213
17.2 Type-2 Fuzzy Logic.. 215
17.3 ANFIS Approach for Building a Type-1 Fuzzy System for Noise Cancellation... 218
17.4 Modified ANFIS Approach for the Type-2 Fuzzy System of Noise Cancellation... 220
17.5 Comparison of Results... 222
17.6 Summary.. 223

Bibliography 225

Appendix 239

Index 241

1 Introduction to Type-2 Fuzzy Logic

We describe in this book, new methods for building intelligent systems using type-2 fuzzy logic and soft computing techniques. Soft Computing (SC) consists of several computing paradigms, including type-1 fuzzy logic, neural networks, and genetic algorithms, which can be used to create powerful hybrid intelligent systems. In this book, we are extending the use of fuzzy logic to a higher order, which is called type-2 fuzzy logic. Combining type-2 fuzzy logic with traditional SC techniques, we can build powerful hybrid intelligent systems that can use the advantages that each technique offers. We consider in this book the use of type-2 fuzzy logic and traditional SC techniques to solve pattern recognition problems in real-world applications. We consider in particular the problems of face, fingerprint and voice recognition. We also consider the problem of recognizing a person by integrating the information given by the face, fingerprint and voice of the person. Other types of applications solved with type-2 fuzzy logic and SC techniques, include intelligent control, intelligent manufacturing, and adaptive noise cancellation.

Fuzzy logic is an area of soft computing that enables a computer system to reason with uncertainty (Castillo & Melin, 2001). A fuzzy inference system consists of a set of if-then rules defined over fuzzy sets. Fuzzy sets generalize the concept of a traditional set by allowing the membership degree to be any value between 0 and 1 (Zadeh, 1965). This corresponds, in the real world, to many situations where it is difficult to decide in an unambiguous manner if something belongs or not to a specific class. Fuzzy expert systems, for example, have been applied with some success to problems of decision, control, diagnosis and classification, just because they can manage the complex expert reasoning involved in these areas of application. The main disadvantage of fuzzy systems is that they can't adapt to changing situations. For this reason, it is a good idea to combine fuzzy logic with neural networks or genetic algorithms, because either one of these last two methodologies could give adaptability to the fuzzy system (Melin & Castillo, 2002). On the other hand, the knowledge that is used to build these fuzzy rules is uncertain. Such uncertainty leads to rules whose antecedents or consequents are uncertain, which translates into uncertain antecedent or consequent membership functions (Karnik & Mendel 1998). Type-1 fuzzy systems, like the ones mentioned above, whose membership functions are type-1 fuzzy sets, are unable to directly handle such uncertainties. We also describe in this book, type-2 fuzzy systems, in which the antecedent or consequent membership functions are

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 1–4, 2008.
springerlink.com © Springer-Verlag Berlin Heidelberg 2008

type-2 fuzzy sets. Such sets are fuzzy sets whose membership grades themselves are type-1 fuzzy sets; they are very useful in circumstances where it is difficult to determine an exact membership function for a fuzzy set. Another way to handle this higher degree of uncertainty is to use intuitionistic fuzzy logic (Atanassov, 1999), which can also be considered as a generalization of type-1 fuzzy logic. In intuitionistic fuzzy logic the uncertainty in describing fuzzy sets is modeled by using at the same time the membership function and the non-membership function of a set (assuming that they are not complementary).

Neural networks are computational models with learning (or adaptive) characteristics that model the human brain (Jang, Sun & Mizutani, 1997). Generally speaking, biological natural neural networks consist of neurons and connections between them, and this is modeled by a graph with nodes and arcs to form the computational neural network. This graph along with a computational algorithm to specify the learning capabilities of the system is what makes the neural network a powerful methodology to simulate intelligent or expert behavior (Miller, Sutton & Werbos, 1995). Neural networks can be classified in supervised and unsupervised. The main difference is that in the case of the supervised neural networks the learning algorithm uses input-output training data to model the dynamic system, on the other hand, in the case of unsupervised neural networks only the input data is given. In the case of an unsupervised network, the input data is used to make representative clusters of all the data. It has been shown, that neural networks are universal approximators, in the sense that they can model any general function to a specified accuracy and for this reason neural networks have been applied to problems of system identification, control, diagnosis, time series prediction, and pattern recognition. We also describe the basic concepts, theory and algorithms of modular and ensemble neural networks. We will also give particular attention to the problem of response integration, which is very important because response integration is responsible for combining all the outputs of the modules. Basically, a modular or ensemble neural network uses several monolithic neural networks to solve a specific problem. The basic idea is that combining the results of several simple neural networks we will achieve a better overall result in terms of accuracy and also learning can be done faster.

Genetic algorithms and evolutionary methods are optimization methodologies based on principles of nature (Jang, Sun & Mizutani, 1997). Both methodologies can also be viewed as searching algorithms because they explore a space using heuristics inspired by nature. Genetic algorithms are based on the ideas of evolution and the biological process that occur at the DNA level. Basically, a genetic algorithm uses a population of individuals, which are modified by using genetic operators in such a way as to eventually obtain the fittest individual (Man, Tang & Kwong, 1999). Any optimization problem has to be represented by using chromosomes, which are a codified representation of the real values of the variables in the problem (Mitchell, 1998). Both, genetic algorithms and evolutionary methods can be used to optimize a general objective function. As genetic algorithms are based on the ideas of natural evolution, we can use this methodology to evolve a neural network or a fuzzy system for a particular application. The problem of finding the best architecture of a neural network is very important because there are no theoretical results on this, and in many cases we are forced to trial and error unless we use a genetic algorithm to automate this process. A similar thing occurs in finding out the optimal number of rules and membership functions of a fuzzy system for a particular application, here a genetic algorithm can also help us avoid time consuming trial and error. In this book, we use genetic algorithms to optimize the architecture of fuzzy and neural systems.

We describe in this book a new approach for face recognition using modular neural networks with a fuzzy logic method for response integration. We describe a new architecture for modular neural networks for achieving pattern recognition in the particular case of human faces. Also, the method for achieving response integration is based on the fuzzy Sugeno integral and type-2 fuzzy logic. Response integration is required to combine the outputs of all the modules in the modular network. We have applied the new approach for face recognition with a real database of faces from students and professors of our institution. Recognition rates with the modular approach were compared against the monolithic single neural network approach, to measure the improvement. The results of the modular neural network approach gives excellent performance overall and also in comparison with the monolithic approach. We also apply this approach for fingerprint recognition using modular neural networks with a fuzzy logic method for response integration. We describe a new architecture for modular neural networks for achieving pattern recognition in the particular case of human fingerprints. Also, the method for achieving response integration is based on the fuzzy Sugeno integral. Response integration is required to combine the outputs of all the modules in the modular network. We have applied the new approach for fingerprint recognition with a real database of fingerprints obtained from students of our institution.

We also describe in this book the use of neural networks, fuzzy logic and genetic algorithms for voice recognition. In particular, we consider the case of speaker recognition by analyzing the sound signals with the help of intelligent techniques, such as the neural networks and fuzzy systems. We use the neural networks for analyzing the sound signal of an unknown speaker, and after this first step, a set of type-2 fuzzy rules is used for decision making. We need to use fuzzy logic due to the uncertainty of the decision process. We also use genetic algorithms to optimize the architecture of the neural networks. We illustrate our approach with a sample of sound signals from real speakers in our institution.

We describe in this book our new approach for human recognition using as information the face, fingerprint, and voice of a person. We have described above the use of intelligent techniques for achieving face recognition, fingerprint recognition, and voice identification. Now we can consider the integration of these three biometric measures to improve the accuracy of human recognition. The new approach will integrate the information from three main modules, one for each of the three biometric measures. The new approach consists in a modular architecture that contains three basic modules: face, fingerprint, and voice. The final decision is based on the results of the three modules and uses type-2 fuzzy logic to take into account the uncertainty of the outputs of the modules.

For the problems of intelligent control of non-linear plants, we also have found that applying type-2 fuzzy logic can help in improving the performance of the controllers. We have considered the design of stable type-2 fuzzy controllers, and the method can be considered a generalization of the fuzzy Lyapunov approach that was already proposed for type-1 fuzzy controllers (Margaliot, 2000). An extensive comparison of the performance of type-2 and type-1 fuzzy controllers has been done under different levels of uncertainty to measure the advantage of type-2 fuzzy logic in this class of problems. The problem of designing type-2 fuzzy controllers using evolutionary computing has also been considered.

Finally, for the problems of intelligent manufacturing, robotics and automation, we have also considered the application of type-2 fuzzy logic and SC techniques for solving this kind of problems. In particular, we have considered the automation of quality control in sound speaker manufacturing by combining interval type-2 fuzzy logic with fractal theory, with excellent results. Plant monitoring and diagnosis is also considered by combining interval type-2 fuzzy logic and fractal theory. Intelligent tracking of autonomous robotic systems is achieved by using a combination of interval type-2 fuzzy logic and genetic algorithms. In this case, the genetic algorithms are used for finding the optimal parameters of the type-2 fuzzy system. Adaptive noise cancellation is also considered by combining interval type-2 fuzzy logic and neural networks. The simulation results show the advantage of using type-2 fuzzy logic for this kind of problems.

2 Type-1 Fuzzy Logic

This chapter introduces the basic concepts, notation, and basic operations for the type-1 fuzzy sets that will be needed in the following chapters. Type-2 fuzzy sets as well as their operations will be discussed in the next chapter. For this reason, in this chapter we will focus only on type-1 fuzzy logic. Since research on fuzzy set theory has been underway for over 30 years now, it is practically impossible to cover all aspects of current developments in this area. Therefore, the main goal of this chapter is to provide an introduction to and a summary of the basic concepts and operations that are relevant to the study of type-1 fuzzy sets. We also introduce in this chapter the definition of linguistic variables and linguistic values and explain how to use them in type-1 fuzzy rules, which are an efficient tool for quantitative modeling of words or sentences in a natural or artificial language. By interpreting fuzzy rules as fuzzy relations, we describe different schemes of fuzzy reasoning, where inference procedures based on the concept of the compositional rule of inference are used to derive conclusions from a set of fuzzy rules and known facts. Fuzzy rules and fuzzy reasoning are the basic components of fuzzy inference systems, which are the most important modeling tool, based on fuzzy set theory.

The "fuzzy inference system" is a popular computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning (Jang, Sun & Mizutani, 1997). It has found successful applications in a wide variety of fields, such as automatic control, data classification, decision analysis, expert systems, time series prediction, robotics, and pattern recognition (Jamshidi, 1997). Because of its multidisciplinary nature, the fuzzy inference system is known by numerous other names, such as "fuzzy expert system" (Kandel, 1992), "fuzzy model" (Sugeno & Kang, 1988), "fuzzy associative memory" (Kosko, 1992), and simply "fuzzy system".

The basic structure of a type-1 fuzzy inference system consists of three conceptual components: a "rule base", which contains a selection of fuzzy rules; a "data base" (or "dictionary"), which defines the membership functions used in the fuzzy rules; and a "reasoning mechanism", which performs the inference procedure upon the rules and given facts to derive a reasonable output or conclusion. In general, we can say that a fuzzy inference system implements a non-linear mapping from its input space to output space. This mapping is accomplished by a number of fuzzy if-then rules, each of

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 5–28, 2008.
springerlink.com © Springer-Verlag Berlin Heidelberg 2008

which describes the local behavior of the mapping. In particular, the antecedent of a rule defines a fuzzy region in the input space, while the consequent specifies the output in the fuzzy region.

We will describe in the following chapter a new area in fuzzy logic, which studies type-2 fuzzy sets and type-2 fuzzy systems. Basically, a type-2 fuzzy set is a set in which we also have uncertainty about the membership function. Since we are dealing with uncertainty for the conventional fuzzy sets (which are called type-1 fuzzy sets here) we can achieve a higher degree of approximation in modeling real world problems. Of course, type-2 fuzzy systems consist of fuzzy if-then rules, which contain type-2 fuzzy sets. We can say that type-2 fuzzy logic is a generalization of conventional fuzzy logic (type-1) in the sense that uncertainty is not only limited to the linguistic variables but also is present in the definition of the membership functions.

In what follows, we shall first introduce the basic concepts of fuzzy sets, and fuzzy reasoning. Then we will introduce and compare the three types of fuzzy inference systems that have been employed in various applications. Finally, we will address briefly the features and problems of fuzzy modeling, which is concerned with the construction of fuzzy inference systems for modeling a given target system. In this chapter, we will assume that all fuzzy sets, fuzzy rules and operations are of type-1 category, unless otherwise specified.

2.1 Type-1 Fuzzy Set Theory

Let X be a space of objects and x be a generic element of X. A classical set A, $A \subseteq X$, is defined by a collection of elements or objects $x \in X$, such that each x can either belong or not belong to the set A. By defining a "characteristic function" for each element $x \in X$, we can represent a classical set A by a set of order pairs (x,0) or (x,1), which indicates $x \notin A$ or $x \in A$, respectively.

Unlike the aforementioned conventional set, a fuzzy set (Zadeh, 1965) expresses the degree to which an element belongs to a set. Hence the characteristic function of a fuzzy set is allowed to have values between 0 and 1, which denotes the degree of membership of an element in a given set.

Definition 2.1. Fuzzy sets and membership functions
If X is a collection of objects denoted generically by x, then a "fuzzy set" A in X is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) \mid x \in X\}. \qquad (2.1)$$

where $\mu_A(x)$ is called "membership function" (or MF for short) for the fuzzy set A. The MF maps each element of X to a membership grade (or membership value) between 0 and 1.

Obviously, the definition of a fuzzy set is a simple extension of the definition of a classical set in which the characteristic function is permitted to have any values between 0 and 1. If the values of the membership function $\mu_A(x)$ is restricted to either 0 or 1, then A is reduced to a classical set and $\mu_A(x)$ is the characteristic function of A. This can be seen with the following example.

Example 2.1. Fuzzy set with a discrete universe of discourse X
Let X = {Tijuana, Acapulco, Cancun} be the set of cities one may choose to organize a conference in. The fuzzy set A = "desirable city to organize a conference in" may be described as follows:

A = {(Tijuana, 0.5), (Acapulco, 0.7), (Cancun, 0.9)}

In this case, the universe of discourse X is discrete- in this example, three cities in Mexico. Of course, the membership grades listed above are quite subjective; anyone can come up with three different values according to his or her preference.

A fuzzy set is uniquely specified by its membership function. To describe membership functions more specifically, we shall define the nomenclature used in the literature (Jang, Sun & Mizutani, 1997).

Definition 2.2. Support
The "support" of a fuzzy set A is the set of all points x in X such that $\mu_A(x) > 0$:

$$\text{support}(A) = \{ x \mid \mu_A(x) > 0 \}. \tag{2.2}$$

Definition 2.3. Core
The "core" of a fuzzy set is the set of all points x in X such that $\mu_A(x) = 1$:

$$\text{core}(A) = \{ x \mid \mu_A(x) = 1 \}. \tag{2.3}$$

Definition 2.4. Normality
A fuzzy set A is "normal" if its core is nonempty. In other words, we can always find a point $x \in X$ such that $\mu_A(x) = 1$.

Definition 2.5. Crossover points
A "crossover point" of a fuzzy set A is a point $x \in X$ at which $\mu_A(x) = 0.5$:

$$\text{crossover}(A) = \{ x \mid \mu_A(x) = 0.5 \}. \tag{2.4}$$

Definition 2.6. Fuzzy singleton
A fuzzy set whose support is a single point in X with $\mu_A(x) = 1$ is called a "fuzzy singleton".

Definition 2.7. α-cut, strong α-cut
The "α-cut" or "α-level set" of a fuzzy set A is a crisp set defined by

$$A_\alpha = \{ x \mid \mu_A(x) \geq \alpha \}. \tag{2.5}$$

"Strong α-cut" or "strong α–level set" are defined similarly:

$$A'_\alpha = \{ x \mid \mu_A(x) > \alpha \}. \tag{2.6}$$

Using the notation for a level set, we can express the support and core of a fuzzy set A as

$$\text{support}(A) = A'_0$$

and

$$\text{core}(A) = A_1$$

respectively.

Corresponding to the ordinary set operations of union, intersection and complement, fuzzy sets have similar operations, which were initially defined in Zadeh's seminal paper (Zadeh, 1965). Before introducing these three fuzzy set operations, first we shall define the notion of containment, which plays a central role in both ordinary and fuzzy sets. This definition of containment is, of course, a natural extension of the case for ordinary sets.

Definition 2.8. Containment

The fuzzy set A is "contained" in fuzzy set B (or, equivalently, A is a "subset" of B) if and only if $\mu_A(x) \leq \mu_B(x)$ for all x. Mathematically,

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x). \tag{2.7}$$

Definition 2.9. Union

The "union" of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cup B$ or C = A OR B, whose MF is related to those of A and B by

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x). \tag{2.8}$$

Definition 2.10. Intersection

The "intersection" of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cap B$ or C = A AND B, whose MF is related to those of A and B by

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x). \tag{2.9}$$

Definition 2.11. Complement or Negation

The "complement" of a fuzzy set A, denoted by A ($\neg$A, NOT A), is defined as

$$\mu_A(x) = 1 - \mu_A(x). \tag{2.10}$$

As mentioned earlier, a fuzzy set is completely characterized by its MF. Since most fuzzy sets in use have a universe of discourse X consisting of the real line R, it would be impractical to list all the pairs defining a membership function. A more convenient and concise way to define a MF is to express it as a mathematical formula. First we define several classes of parameterized MFs of one dimension.

Definition 2.12. Triangular MFs

A "triangular MF" is specified by three parameters {a, b, c} as follows:

$$y = \text{triangle}(x;a,b,c) = \begin{cases} 0, & x \leq a. \\ (x-a)/(b-a), & a \leq x \leq b. \\ (c-x)/(c-b), & b \leq x \leq c. \\ 0, & c \leq x. \end{cases} \tag{2.11}$$

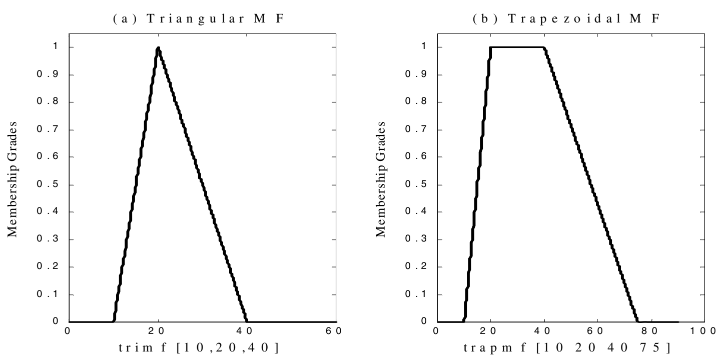
The parameters {a,b,c} (with a < b < c) determine the x coordinates of the three corners of the underlying triangular MF. Figure 2.1 (a) illustrates a triangular MF defined by triangle(x; 10, 20, 40).

Definition 2.13. Trapezoidal MFs
A "trapezoidal MF" is specified by four parameters {a, b, c, d} as follows:

$$\text{trapezoid}(x;a,b,c,d) = \begin{cases} 0, & x \le a. \\ (x-a)/(b-a), & a \le x \le b. \\ 1, & b \le x \le c. \\ (d-x)/(d-c), & c \le x \le d. \\ 0, & d \le x. \end{cases} \quad (2.12)$$

The parameters {a, b, c, d} (with a < b ≤ c <d) determine the x coordinates of the four corners of the underlying trapezoidal MF. Figure 2.1 (b) illustrates a trapezoidal MF defined by trapezoid(x; 10, 20 40, 75).

Due to their simple formulas and computational efficiency, both triangular MFs and trapezoidal MFs have been used extensively, especially in real-time implementations. However, since the MFs are composed of straight line segments, they are not smooth at the corner points specified by the parameters. In the following we introduce other types of MFs defined by smooth and nonlinear functions.



(a) Triangular MF b) Trapezoidal MF

Fig. 2.1. Examples of two types of parameterized MFs

Definition 2.14. Gaussian MFs
A "Gaussian MF" is specified by two parameters {c , σ }:

$$\text{gaussian}(x; c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}. \quad (2.13)$$

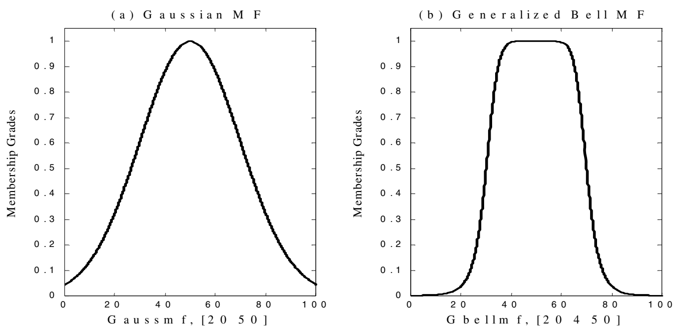
A “Gaussian” MF is determined completely by c and σ ; c represents the MFs center and σ determines the MFs width. Figure 2.2 (a) plots a Gaussian MF defined by gaussian (x; 50, 20).

Definition 2.15. Generalized bell MFs
A "generalized bell MF" is specified by three parameters {a, b, c}:

$$\text{bell}(x; a, b, c) = \frac{1}{1 + |(x-c)/a|^{2b}} \tag{2.14}$$

where the parameter b is usually positive. We can note that this MF is a direct generalization of the Cauchy distribution used in probability theory, so it is also referred to as the "Cauchy MF". Figure 2.2 (b) illustrates a generalized bell MF defined by bell(x; 20, 4, 50).

Although the Gaussian MFs and bell MFs achieve smoothness, they are unable to specify asymmetric MFs, which are important in certain applications. Next we define the sigmoidal MF, which is either open left or right.



(a) Gaussian MF (b) Generalized Bell MF

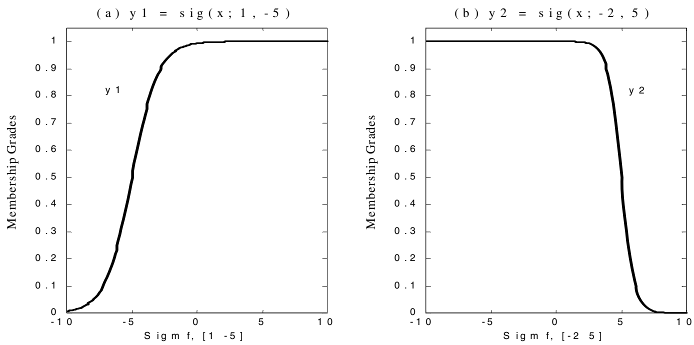
Fig. 2.2. Examples of two classes of parameterized continuous MFs

Definition 2.16. Sigmoidal MFs
A "Sigmoidal MF" is defined by the following equation:

$$\text{sig}(x; a, c) = \frac{1}{1 + \exp[-a(x-c)]} \tag{2.15}$$

where a controls the slope at the crossover point x = c.

Depending on the sign of the parameter "a", a sigmoidal MF is inherently open right or left and thus is appropriate for representing concepts such as "very large" or "very negative". Figure 2.3 shows two sigmoidal functions y_1 =sig(x; 1, -5) and y_2 =sig(x; -2, 5).



(a) y_1 = sig(x; 1, -5) (b) y_2 = sig(x; -2, 5)

Fig. 2.3. Two sigmoidal functions y_1 and y_2

2.2 Fuzzy Rules and Fuzzy Reasoning

In this section we introduce the concepts of the extension principle and fuzzy relations, which extend the notions of fuzzy sets introduced previously. Then we give the definition of linguistic variables and linguistic values and show how to use them in fuzzy rules. By interpreting fuzzy rules as fuzzy relations, we describe different schemes of fuzzy reasoning. Fuzzy rules and fuzzy reasoning are the backbone of fuzzy inference systems, which are the most important modeling tool based on fuzzy set theory.

2.2.1 Fuzzy Relations

The "extension principle" is a basic concept of fuzzy set theory that provides a general procedure for extending crisp domains of mathematical expressions to fuzzy domains. This procedure generalizes a common one-to-one mapping of a function f to a mapping between fuzzy sets. More specifically, lets assume that f is a function from X to Y and A is a fuzzy set on X defined as

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \ldots + \mu_A(x_n)/x_n$$

Then the extension principle states that the image of fuzzy set A under the mapping f can be expressed as a fuzzy set B,

$$B = f(A) = \mu_A(x_1)/y_1 + \mu_A(x_2)/y_2 + \ldots + \mu_A(x_n)/y_n$$

where $y_i = f(x_i)$, i = 1, …, n. In other words, the fuzzy set B can be defined through the values of f in x_1, x_2, …, x_n. If f is a many-to-one mapping, then there exists x_1, x_2 $\in$ X, $x_1 \neq x_2$, such that $f(x_1) = f(x_2) = y^*$, $y^* \in$ Y. In this case, the membership grade of B at y = y* is the maximum of the membership grades of A at $x = x_1$ and $x = x_2$, since f(x) = y* may result from $x = x_1$ or $x = x_2$. More generally speaking, we have

$$\mu_B(y) = \max_{x = f^{-1}(y)} \mu_A(x).$$

A simple example of this concept is shown below.

Example 2.2. Application of the extension principle to fuzzy sets
Let suppose we have the following fuzzy set with discrete universe

$$A = 0.2/ -2 + 0.5/ -1 + 0.7/ 0 + 0.9/ 1 + 0.4/ 2$$

and lets suppose that we have the following mapping

$$y = x^2 + 1.$$

After applying the extension principle, we have the following result

$$B = 0.2/ 5 + 0.5/ 2 + 0.7/ 1 + 0.9/ 2 + 0.4/ 5$$
$$B = 0.7/ 1 + (0.2 \vee 0.4)/ 5 + (0.5 \vee 0.9)/ 2$$
$$B = 0.7/ 1 + 0.4/ 5 + 0.9/ 2,$$

where $\vee$ represents "max".

Binary fuzzy relations are fuzzy sets in $X \times Y$ which map each element in $X \times Y$ to a membership grade between 0 and 1. In particular, unary fuzzy relations are fuzzy sets with one-dimensional MFs; binary fuzzy relations are fuzzy sets with two-dimensional MFs, and so on. Here we will restrict our attention to binary fuzzy relations. A generalization to n-ary fuzzy relations is not so difficult.

Definition 2.17. Binary fuzzy relation
Let X and Y be two universes of discourse. Then

$$\Re = \{ ((x, y), \mu_{\Re}(x, y)) \mid (x, y) \in X \times Y\} \tag{2.16}$$

is a binary fuzzy relation in $X \times Y$.

Example 2.3. Binary fuzzy relations
Let X = {1, 2, 3} and Y = {1, 2, 3, 4, 5} and $\Re$ = "y is slightly greater than x". The MF of the fuzzy relation $\Re$ can be defined (subjectively) as

$$\mu_{\Re}(x, y) = \begin{cases} (y - x)/(y + x), \text{ if } y > x. \\ 0, \text{ if } y \le x. \end{cases} \tag{2.17}$$

This fuzzy relation $\Re$ can be expressed as a relation matrix in the following form:

$$\Re = \begin{pmatrix} 0 & 0.333 & 0.500 & 0.600 & 0.666 \\ 0 & 0 & 0.200 & 0.333 & 0.428 \\ 0 & 0 & 0 & 0.142 & 0.250 \end{pmatrix}$$

where the element at row i and column j is equal to the membership grade between the *i*th element of X and *j*th element of Y.

Other common examples of binary fuzzy relations are the following:

- x is similar to y (x and y are objects)
- x depends on y (x and y are events)
- If x is big, then y is small (x is an observed reading and y is the corresponding action)

The last example, “If x is A, then y is B”, is used repeatedly in fuzzy systems. We will explore fuzzy relations of this type in the following section.

Fuzzy relations in different product spaces can be combined through a composition operation. Different composition operations have been proposed for fuzzy relations; the best known is the max-min composition proposed by Zadeh in 1965.

Definition 2.18. Max-min composition
Let $\Re_1$ and $\Re_2$ be two fuzzy relations defined on $X \times Y$ and $Y \times Z$, respectively. The “max-min composition” of $\Re_1$ and $\Re_2$ is a fuzzy set defined by

$$\Re_1 o \Re_2 = \{[(x, z), \max_y \min(\mu_{\Re 1}(x, y), \mu_{\Re 2}(y, z))] | x \in X, y \in Y, z \in Z\} \tag{2.18}$$

When $\Re_1$ and $\Re_2$ are expressed as relation matrices, the calculation of the composition $\Re_1 o \Re_2$ is almost the same as matrix multiplication, except that $\times$ and + are replaced by the “min” and “max” operations, respectively. For this reason, the max-min composition is also called the “max-min product”.

2.2.2 Fuzzy Rules

As was pointed out by Zadeh in his work on this area (Zadeh, 1973), conventional techniques for system analysis are intrinsically unsuited for dealing with humanistic systems, whose behavior is strongly influenced by human judgment, perception, and emotions. This is a manifestation of what might be called the "principle of incompatibility": "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 become almost mutually exclusive characteristics" (Zadeh, 1973). It was because of this belief that Zadeh proposed the concept of linguistic variables (Zadeh, 1971) as an alternative approach to modeling human thinking.

Definition 2.19. Linguistic variables
A "Linguistic variable" is characterized by a quintuple (x, T(x), X, G, M) in which x is the name of the variable; T(x) is the "term set" of x-that is, the set of its "linguistic values" or "linguistic terms"; X is the universe of discourse, G is a "syntactic rule" which generates the terms in T(x); and M is a "semantic rule" which associates with each linguistic value A its meaning M(A), where M(A) denotes a fuzzy set in X.

Definition 2.20. Concentration and dilation of linguistic values
Let A be a linguistic value characterized by a fuzzy set membership function $\mu_A(.)$. Then A^k is interpreted as a modified version of the original linguistic value expressed as

$$A^k = \int_X [\mu_A(x)]^k / x \ . \tag{2.19}$$

In particular, the operation of "concentration" is defined as

$$CON(A) = A^2 \ , \tag{2.20}$$

while that of "dilation" is expressed by

$$DIL(A) = A^{0.5} \quad . \tag{2.21}$$

Conventionally, we take CON(A) and DIL(A) to be the results of applying the hedges "very" and "more or less", respectively, to the linguistic term A. However, other consistent definitions for these linguistic hedges are possible and well justified for various applications.

Following the definitions given before, we can interpret the negation operator NOT and the connectives AND and OR as

$$\begin{aligned} NOT(A) &= \neg A = \int_X [\,1 - \mu_A(x)\,] / x \quad , \\ A \text{ AND } B &= A \cap B = \int_X [\,\mu_A(x) \wedge \mu_B(x)\,] / x \, , \\ A \text{ OR } B &= A \cup B = \int_X [\,\mu_A(x) \vee \mu_B(x)\,] / x \quad . \end{aligned} \tag{2.22}$$

respectively, where A and B are two linguistic values whose meanings are defined by $\mu_A(.)$ and $\mu_B(.)$.

Definition 2.21. Fuzzy If-Then Rules

A "fuzzy if-then rule" (also known as "fuzzy rule", "fuzzy implication", or "fuzzy conditional statement") assumes the form

$$\textbf{if } x \text{ is } A \textbf{ then } y \text{ is } B \quad , \tag{2.23}$$

where A and B are linguistic values defined by fuzzy sets on universes of discourse X and Y, respectively. Often "x is A" is called "antecedent" or "premise", while "y is B" is called the "consequence" or "conclusion".

Examples of fuzzy if-then rules are widespread in our daily linguistic expressions, such as the following:

- If pressure is high, then volume is small.
- If the road is slippery, then driving is dangerous.
- If the speed is high, then apply the brake a little.

Before we can employ fuzzy if-then rules to model and analyze a system, first we have to formalize what is meant by the expression "if x is A then y is B", which is sometimes abbreviated as $A \rightarrow B$. In essence, the expression describes a relation between two variables x and y; this suggests that a fuzzy if-then rule is defined as a binary fuzzy relation R on the product space $X \times Y$. Generally speaking, there are two ways to interpret the fuzzy rule $A \rightarrow B$. If we interpret $A \rightarrow B$ as A "coupled with" B then

$$R = A \rightarrow B = A \times B = \int_{X \times Y} \mu_A(x) * \mu_B(y) / (x,y)$$

where * is an operator for intersection (Mamdani & Assilian, 1975). On the other hand, if $A \rightarrow B$ is interpreted as A "entails" B, then it can be written as one of two different formulas:

- Material implication:

$$R = A \rightarrow B = \neg A \cup B \quad . \tag{2.24}$$

- Propositional Calculus:

$$R = A \rightarrow B = \neg A \cup (A \cap B) \quad . \tag{2.25}$$

Although these two formulas are different in appearance, they both reduce to the familiar identity $A \rightarrow B \equiv \neg A \cup B$ when A and B are propositions in the sense of two-valued logic.

Fuzzy reasoning, also known as approximate reasoning, is an inference procedure that derives conclusions from a set of fuzzy if-then rules and known facts. The basic rule of inference in traditional two-valued logic is "modus ponens", according to which we can infer the truth of a proposition B from the truth of A and the implication $A \rightarrow B$. This concept is illustrated as follows:

premise 1 (fact):	x is A ,
premise 2 (rule):	if x is A then y is B ,
consequence (conclusion):	y is B .

However, in much of human reasoning, modus ponens is employed in an approximate manner. This is written as

premise 1 (fact):	x is A'
premise 2 (rule):	if x is A then y is B ,
consequence (conclusion):	y is B'

where A' is close to A and B' is close to B. When A, B, A' and B' are fuzzy sets of appropriate universes, the foregoing inference procedure is called "approximate reasoning" or "fuzzy reasoning"; it is also called "generalized modus ponens" (GMP for short), since it has modus ponens as a special case.

Definition 2.22. Fuzzy reasoning

Let A, A', and B be fuzzy sets of X, X, and Y respectively. Assume that the fuzzy implication $A \rightarrow B$ is expressed as a fuzzy relation R on $X \times Y$. Then the fuzzy set B induced by "x is A'" and the fuzzy rule "if x is A then y is B" is defined by

$$\begin{aligned} \mu_{B'}(y) &= \max_X \min [\, \mu_{A'}(x), \mu_R(x, y) \,] \\ &= \vee_X [\, \mu_{A'}(x) \wedge \mu_R(x, y) \,] \, . \end{aligned} \tag{2.26}$$

Now we can use the inference procedure of fuzzy reasoning to derive conclusions provided that the fuzzy implication $A \rightarrow B$ is defined as an appropriate binary fuzzy relation.

Single Rule with Single Antecedent

This is the simplest case, and the formula is available in Equation (2.26). A further simplification of the equation yields

$$\begin{aligned} \mu_{B'}(y) &= [\, \vee_X (\, \mu_{A'}(x) \wedge \mu_A(x) \,) \,] \wedge \mu_B(y) \\ &= \omega \wedge \mu_B(y) \end{aligned}$$

In other words, first we find the degree of match ω as the maximum of $\mu_{A'}(x) \wedge \mu_A(x)$; then the MF of the resulting B' is equal to the MF of B clipped by ω. Intuitively, ω represents a measure of degree of belief for the antecedent part of a rule; this measure gets propagated by the if-then rules and the resulting degree of belief or MF for the consequent part should be no greater than ω.

Multiple Rules with Multiple Antecedents

The process of fuzzy reasoning or approximate reasoning for the general case can be divided into four steps:

1) **Degrees of compatibility:** Compare the known facts with the antecedents of fuzzy rules to find the degrees of compatibility with respect to each antecedent MF.
2) **Firing Strength:** Combine degrees of compatibility with respect to antecedent MFs in a rule using fuzzy AND or OR operators to form a firing strength that indicates the degree to which the antecedent part of the rule is satisfied.
3) **Qualified (induced) consequent MFs:** Apply the firing strength to the consequent MF of a rule to generate a qualified consequent MF.
4) **Overall output MF:** Aggregate all the qualified consequent MFs to obtain an overall output MF.

2.3 Fuzzy Inference Systems

In this section we describe the three types of fuzzy inference systems that have been widely used in the applications. The differences between these three fuzzy inference systems lie in the consequents of their fuzzy rules, and thus their aggregation and defuzzification procedures differ accordingly.

The "Mamdani fuzzy inference system" (Mamdani & Assilian, 1975) was proposed as the first attempt to control a steam engine and boiler combination by a set of linguistic control rules obtained from experienced human operators. Figure 2.4 is an illustration of how a two-rule Mamdani fuzzy inference system derives the overall output z when subjected to two numeric inputs x and y.

In Mamdani's application, two fuzzy inference systems were used as two controllers to generate the heat input to the boiler and throttle opening of the engine cylinder, respectively, to regulate the steam pressure in the boiler and the speed of the engine. Since the engine and boiler take only numeric values as inputs, a defuzzifier was used to convert a fuzzy set to a numeric value.

Defuzzification

Defuzzification refers to the way a numeric value is extracted from a fuzzy set as a representative value. In general, there are five methods for defuzzifying a fuzzy set A of a universe of discourse Z, as shown in Figure 2.5 (Here the fuzzy set A is usually represented by an aggregated output MF, such as C' in Figure 2.4). A brief explanation of each defuzzification strategy follows.

- Centroid of area z_{COA}:

$$z_{COA} = \frac{\int_Z \mu_A(z) z dz}{\int_Z \mu_A(z) dz} \qquad (2.27)$$

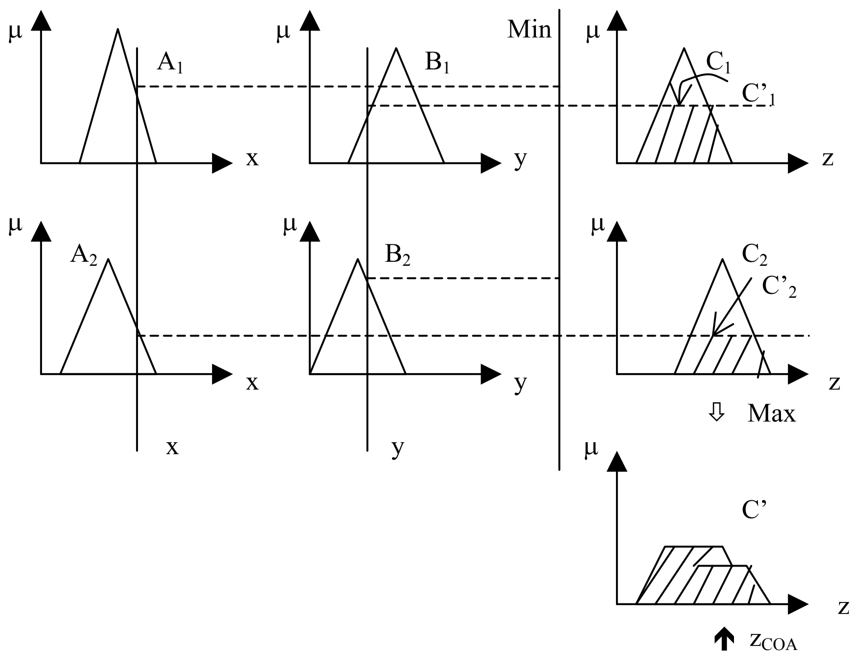


Fig. 2.4. The Mamdani fuzzy inference system using the min and max operators

where $\mu_A(z)$ is the aggregated output MF. This is the most widely adopted defuzzification strategy, which is reminiscent of the calculation of expected values of probability distributions.

- Bisector of area z_{BOA} : z_{BOA} satisfies

$$\int_{\alpha}^{z_{BOA}} \mu_A(z)dz = \int_{z_{BOA}}^{\beta} \mu_A(z)dz \quad , \tag{2.28}$$

where $\alpha = \min\{z \mid z \in Z\}$ and $\beta = \max\{z \mid z \in Z\}$.

- Mean of maximum z_{MOM} : z_{MOM} is the average of the maximizing z at which the MF reach a maximum μ^*. Mathematically,

$$z_{MOM} = \frac{\int_{z'} z dz}{\int_{z'} dz} \quad , \tag{2.29}$$

where $z' = \{ z \mid \mu_A(z) = \mu^* \}$. In particular, if $\mu_A(z)$ has a single maximum at $z = z^*$, then $z_{MOM} = z^*$. Moreover, if $\mu_A(z)$ reaches its maximum whenever $z \in [z_{left}, z_{right}]$ then $z_{MOM} = (z_{left} + z_{right}) / 2$.

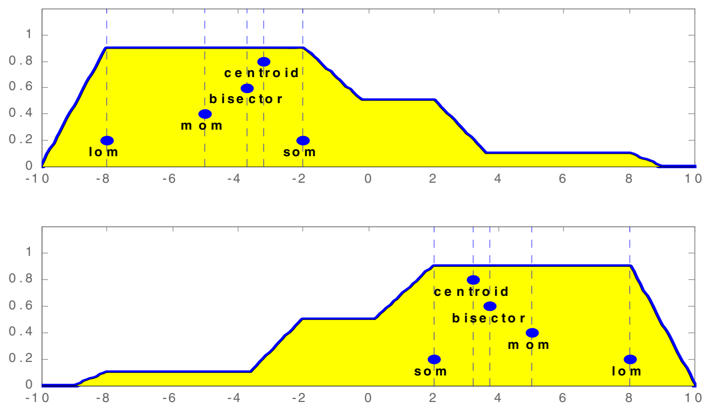


Fig. 2.5. Various defuzzification methods for obtaining a numeric output

- Smallest of maximum z_{SOM} : z_{SOM} is the minimum (in terms of magnitude) of the maximizing z.
- Largest of maximum z_{LOM} : z_{LOM} is the maximum (in terms of magnitude) of the maximizing z. Because of their obvious bias, z_{SOM} and z_{LOM} are not used as often as the other three defuzzification methods.

The calculation needed to carry out any of these five defuzzification operations is time-consuming unless special hardware support is available. Furthermore, these defuzzification operations are not easily subject to rigorous mathematical analysis, so most of the studies are based on experimental results. This leads to the propositions of other types of fuzzy inference systems that do not need defuzzification at all; two of them will be described in the following. Other more flexible defuzzification methods can be found in several more recent papers (Yager & Filev, 1993), (Runkler & Glesner, 1994).

We will give a simple example to illustrate the use of the Mamdani fuzzy inference system. We will consider the case of determining the quality of a image produce by a Television as a result of controlling the electrical tuning process based on the input variables: voltage and current. Automating the electrical tuning process during the manufacturing of televisions, results in increased productivity and reduction of production costs, as well as increasing the quality of the imaging system of the television. The fuzzy model will consist of a set of rules relating these variables, which represent expert knowledge in the electrical tuning process of televisions. In Figure 2.6 we show the architecture of the fuzzy system relating the input variables (voltage, current and time) with the output variable (quality of the image), which was implemented by using the MATLAB Fuzzy Logic Toolbox. We show in Figure 2.7 the fuzzy rule base, which was implemented by using the “rule editor” of the same toolbox. In Figure 2.8 we can appreciate the membership functions for the image-quality variable. We show in Figure 2.9 the membership functions for the voltage variable. We also show in Figure 2.10 the use of the “rule viewer” of MATLAB to calculate specific values. Finally, in Figure 2.11 we show the non-linear surface for the Mamdani model.

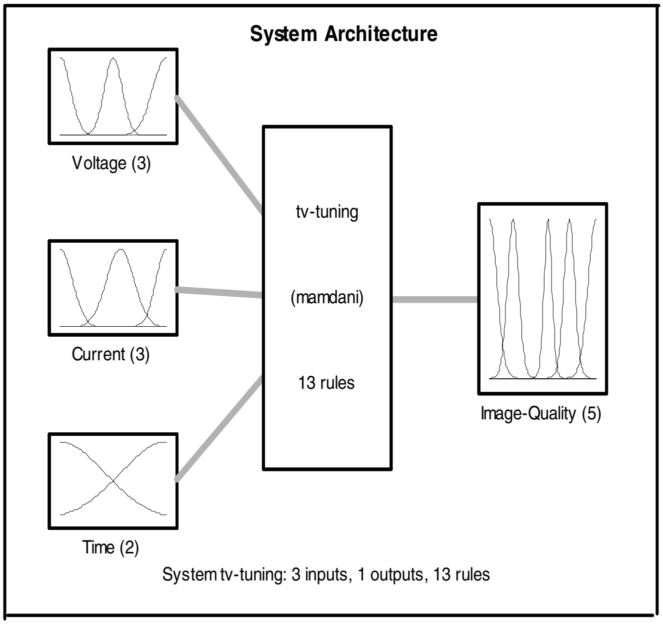


Fig. 2.6. Architecture of the fuzzy system for quality evaluation

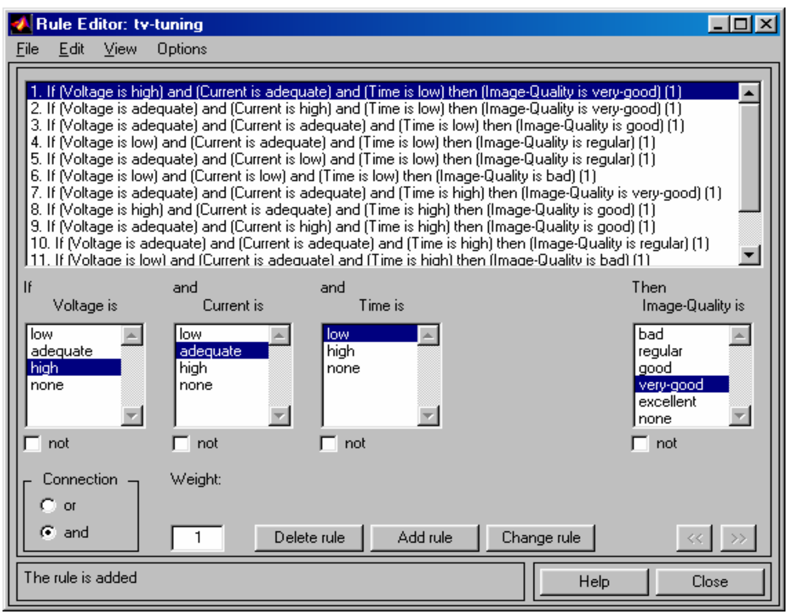


Fig. 2.7. Fuzzy rule base for quality evaluation

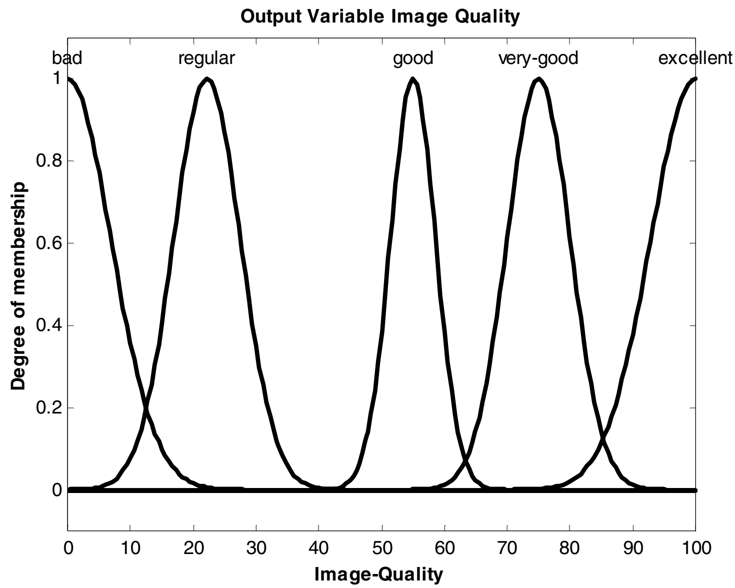


Fig. 2.8. Gaussian membership functions for the output linguistic variable

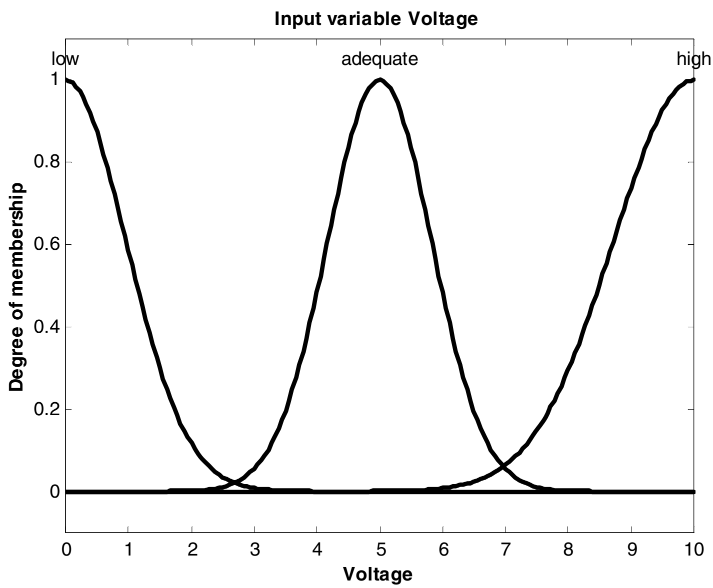


Fig. 2.9. Gaussian membership functions for the voltage linguistic variable

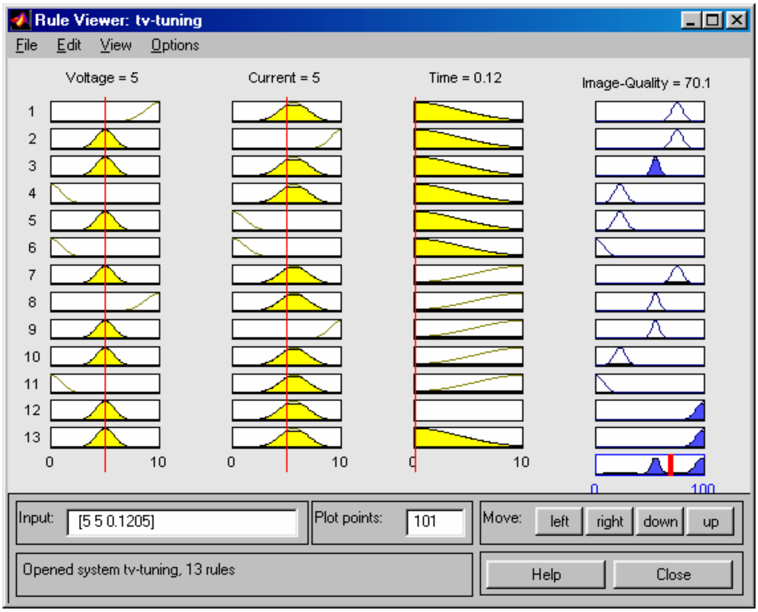


Fig. 2.10. Use of the fuzzy rule base with specific values

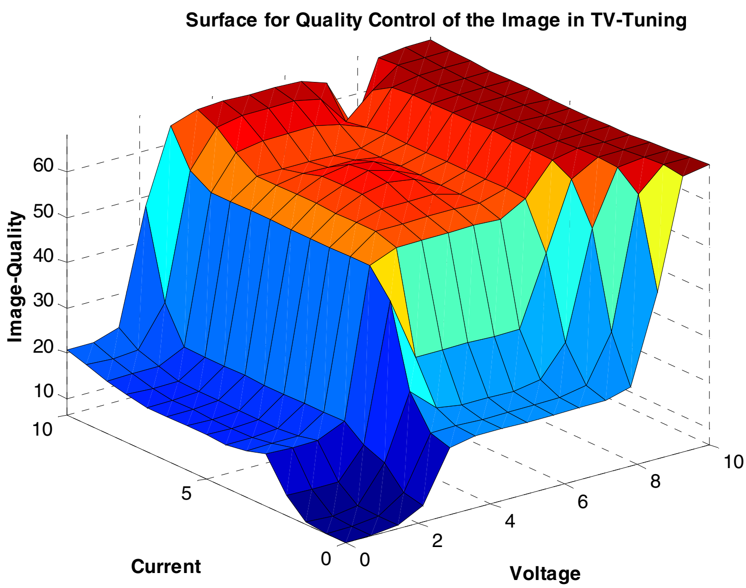


Fig. 2.11. Non-linear surface of the Mamdani fuzzy model

Sugeno Fuzzy Models

The "Sugeno fuzzy model" (also known as the "TSK fuzzy model") was proposed by Takagi, Sugeno and Kang in an effort to develop a systematic approach to generating fuzzy rules from a given input-output data set (Takagi & Sugeno, 1985), (Sugeno & Kang, 1988). A typical fuzzy rule in a Sugeno fuzzy model has the form:

if x is A **and** y is B **then** z = f(x,y)

where A and B are fuzzy sets in the antecedent, while z = f(x,y) is a traditional function in the consequent. Usually f(x,y) is a polynomial in the input variables x and y, but it can be any function as long as it can appropriately describe the output of the model within the fuzzy region specified by the antecedent of the rule. When f(x,y) is a first-order polynomial, the resulting fuzzy inference system is called a "first-order Sugeno fuzzy model". When f is constant, we then have a "zero-order Sugeno fuzzy model", which can be viewed either as a special case of the Mamdani inference system, in which each rule's consequent is specified by a fuzzy singleton, or a special case of the Tsukamoto fuzzy model (to be introduced next), in which each rule's consequent is specified by a MF of a step function center at the constant.

Figure 2.12 shows the fuzzy reasoning procedure for a first-order Sugeno model. Since each rule has a numeric output, the overall output is obtained via "weighted average", thus avoiding the time-consuming process of defuzzification required in a Mamdani model. In practice, the weighted average operator is sometimes replaced with the "weighted sum" operator (that is, $w_1z_1 + w_2z_2$ in Figure 2.12) to reduce computation further specially, in the training of a fuzzy inference system. However, this simplification could lead to the loss of MF linguistic meanings unless the sum of firing strengths (that is, $\sum w_i$) is close to unity.

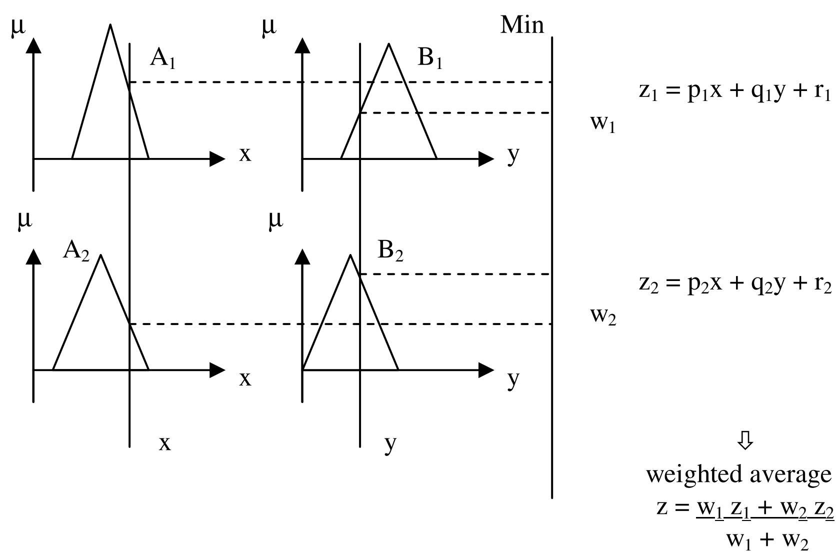


Fig. 2.12. The Sugeno fuzzy model

Unlike the Mamdani fuzzy model, the Sugeno fuzzy model cannot follow the compositional rule of inference strictly in its fuzzy reasoning mechanism. This poses some difficulties when the inputs to a Sugeno fuzzy model are fuzzy. Specifically, we can still employ the matching of fuzzy sets to find the firing strength of each rule. However, the resulting overall output via either weighted average or weighted sum is always crisp; this is counterintuitive since a fuzzy model should be able to propagate the fuzziness from inputs to outputs in an appropriate manner. Without the use of the time-consuming defuzzification procedure, the Sugeno fuzzy model is by far the most popular candidate for sample-data-based modeling.

We will give a simple example to illustrate the use of the Sugeno fuzzy inference system. We will consider again the television example, i.e., determining the quality of the images produced by the television depending on the voltage and current of the electrical tuning process. In Figure 2.13 we show the architecture of the Sugeno model for this example. We show in Figure 2.14 the fuzzy rule base of the Sugeno model. We also show in Figure 2.15 the membership functions for the current input variable. In Figure 2.16 we show the non-linear surface of the Sugeno model.

Finally, we show in Figure 2.17 the use of the "rule viewer" of the Fuzzy Logic Toolbox of MATLAB. The rule viewer is used when we want to evaluate the output of a fuzzy system using specific values for the input variables. In Figure 2.17, for example, we give a voltage of 5 volts, a current intensity of 5 Amperes, and a time of production of 5 seconds, and obtain as a result a quality of 92.2%, which is excellent. Of course, this is only an illustrative example of the potential use of fuzzy logic in this type of applications.

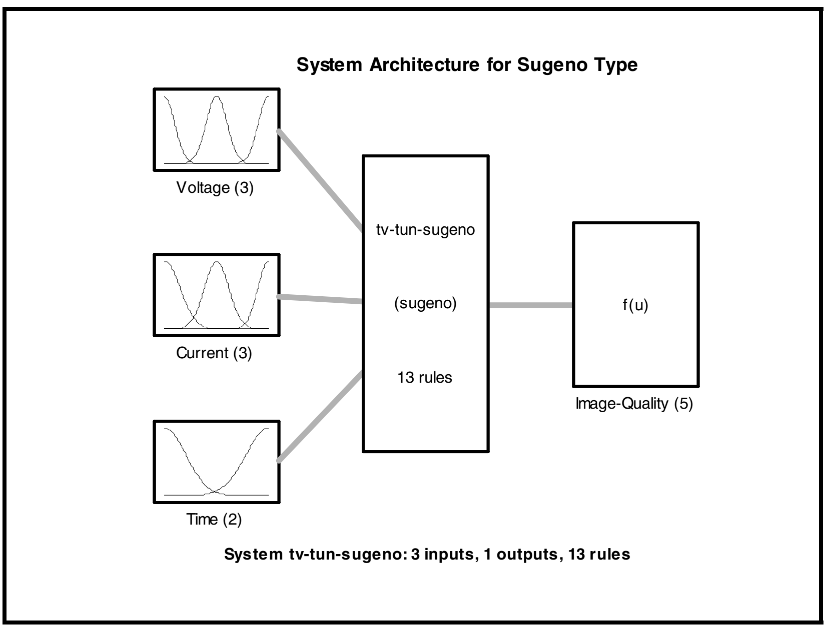


Fig. 2.13. Architecture of the Sugeno fuzzy model for quality evaluation

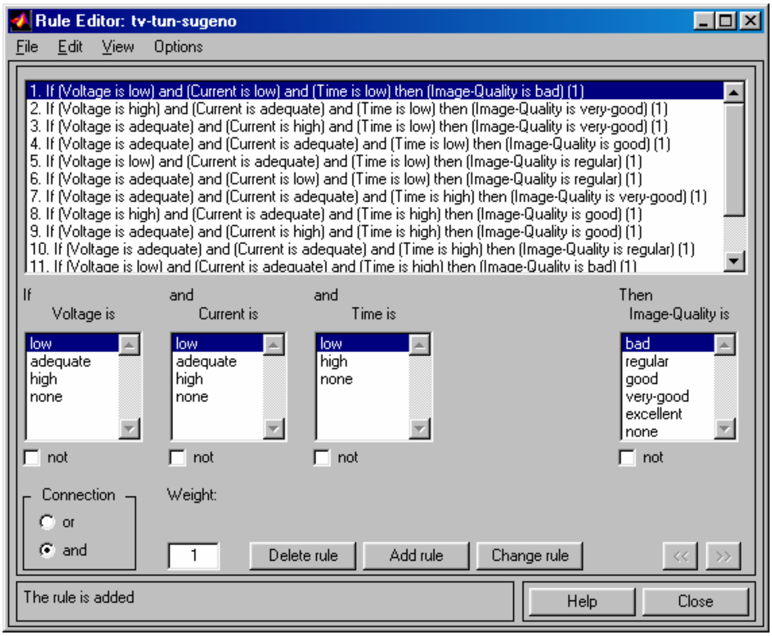


Fig. 2.14. Fuzzy rule base for quality evaluation using the "rule editor"

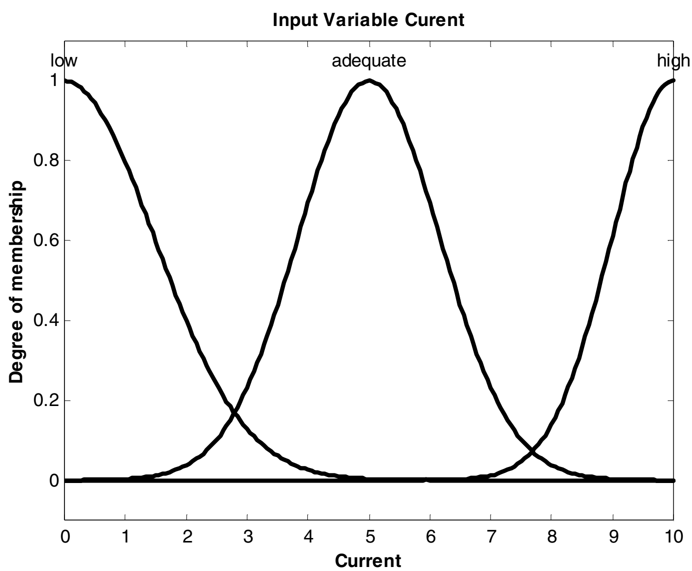


Fig. 2.15. Membership functions for the current linguistic variable

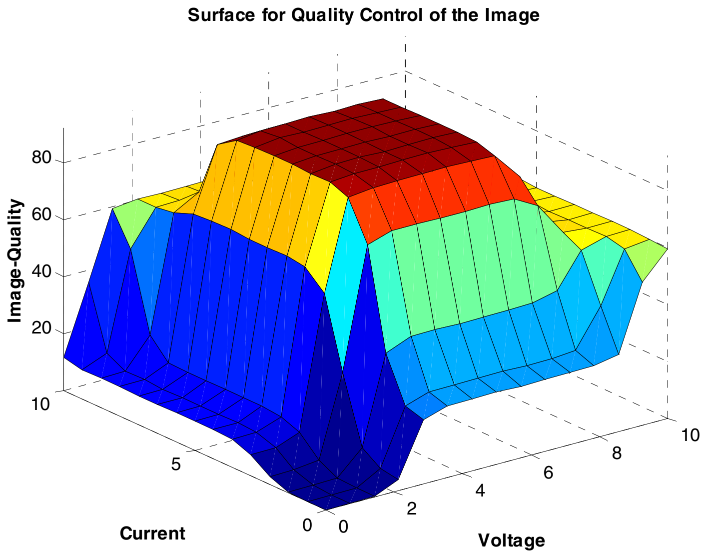


Fig. 2.16. Non-linear surface for the Sugeno fuzzy model for quality evaluation

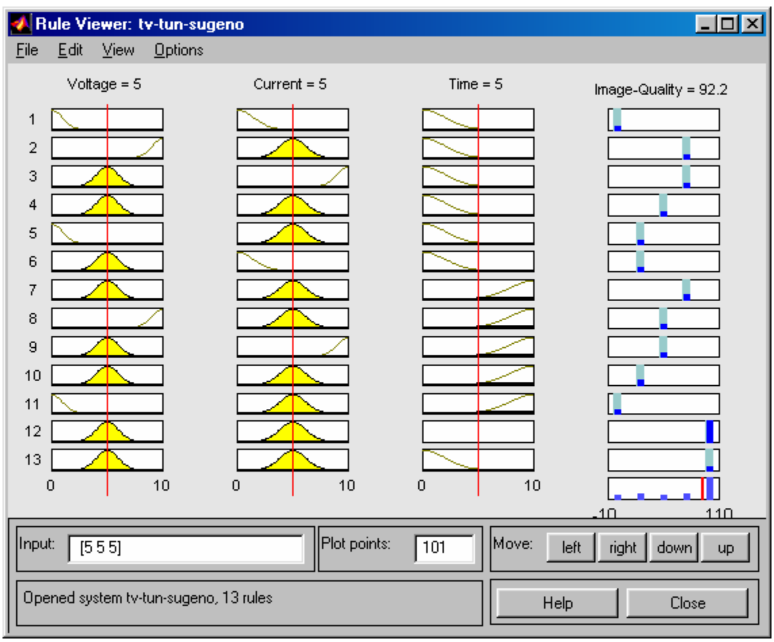


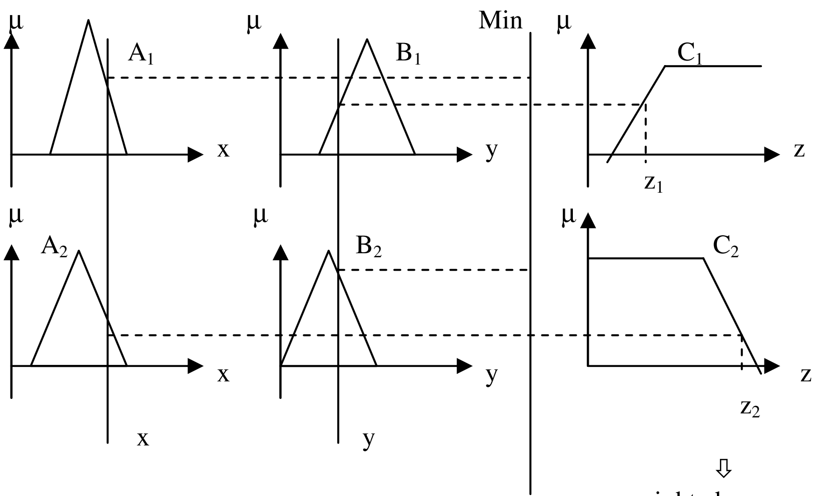
Fig. 2.17. Application of the rule viewer of MATLAB with specific values

Tsukamoto Fuzzy Models

In the "Tsukamoto fuzzy models" (Tsukamoto, 1979), the consequent of each fuzzy if-then rule is represented by a fuzzy set with a monotonical MF, as shown in

Figure 2.18. As a result, the inferred output of each rule is defined as a numeric value induced by the rule firing strength. The overall output is taken as the weighted average of each rule's output. Figure 2.18 illustrates the reasoning procedure for a two-input two-rule system.

Since each rule infers a numeric output, the Tsukamoto fuzzy model aggregates each rule's output by the method of weighted average and thus avoids the time-consuming process of defuzzification. However, the Tsukamoto fuzzy model is not used often since it is not as transparent as either the Mamdani or Sugeno fuzzy models. Since the reasoning method of the Tsukamoto fuzzy model does not follow strictly the compositional rule of inference, the output is always crisp even when the inputs are fuzzy.



$$z = \frac{w_1 z_1 + w_2 z_2}{w_1 + w_2}$$

Fig. 2.18. The Tsukamoto fuzzy model

There are certain common issues concerning all the three fuzzy inference systems introduced previously, such as how to partition an input space and how to construct a fuzzy inference system for a particular application. We will examine these issues in more detail in the following lines.

Input Space Partitioning

Now it should be clear that the main idea of fuzzy inference systems resembles that of "divide and conquer" - the antecedent of a fuzzy rule defines a local fuzzy region, while the consequent describes the behavior within the region via various constituents. The consequent constituent can be a consequent MF (Mamdani and Tsukamoto fuzzy models), a constant value (zero-order Sugeno model), a linear equation (first-order Sugeno model) or a non-linear equation (higher order Sugeno models). Different consequent constituents result in different fuzzy inference systems, but their antecedents are always the same. Therefore, the following discussion of methods of partitioning input spaces to form the antecedents of fuzzy rules is applicable to all three types of fuzzy inference systems.

- Grid partition: This partition method is often chosen in designing a fuzzy controller, which usually involves only several state variables as the inputs to the controller. This partition strategy needs only a small number of MFs for each input. However, it encounters problems when we have a moderately large number of inputs. For instance, a fuzzy model with 12 inputs and 2 MFs on each input would result in $2^{12} = 4096$ fuzzy if-then rules, which is prohibitively large. This problem, usually referred to as the "curse of dimensionality", can be alleviated by other partition strategies.
- Tree partition: In this method each region can be uniquely specified along a corresponding decision tree. The tree partition relieves the problem of an exponential increase in the number of rules. However, more MFs for each input are needed to define these fuzzy regions, and these MFs do not usually bear clear linguistic meanings. In other words, orthogonality holds roughly in $X \times Y$, but not in either X or Y alone.
- Scatter partition: By covering a subset of the whole input space that characterizes a region of possible occurrence of the input vectors, the scatter partition can also limit the number of rules to a reasonable amount. However, the scatter partition is usually dictated by desired input-output data pairs and thus, in general, orthogonality does not hold in X, Y or $X \times Y$. This makes it hard to estimate the overall mapping directly from the consequent of each rule's output.

2.4 Fuzzy Modeling

In general, we design a fuzzy inference system based on the past known behavior of a target system. The fuzzy system is then expected to be able to reproduce the behavior of the target system. For example, if the target system is a human operator in charge of a electrochemical reaction process, then the fuzzy inference system becomes a fuzzy logic controller that can regulate and control the process.

Let us now consider how we might construct a fuzzy inference system for a specific application. Generally speaking, the standard method for constructing a fuzzy inference system, a process usually called "fuzzy modeling", has the following features:

- The rule structure of a fuzzy inference system makes it easy to incorporate human expertise about the target system directly into the modeling process. Namely, fuzzy modeling takes advantage of "domain knowledge" that might not be easily or directly employed in other modeling approaches.
- When the input-output data of a target system is available, conventional system identification techniques can be used for fuzzy modeling. In other words, the use of "numerical data" also plays an important role in "fuzzy modeling", just as in other mathematical modeling methods.

Conceptually, fuzzy modeling can be pursued in two stages, which are not totally disjoint. The first stage is the identification of the "surface structure", which includes the following tasks:

1. Select relevant input and output variables.
2. Choose a specific type of fuzzy inference system.
3. Determine the number of linguistic terms associated with each input and output variables.
4. Design a collection of fuzzy if-then rules.

Note that to accomplish the preceding tasks, we rely on our own knowledge (common sense, simple physical laws, an so on) of the target system, information provided by human experts who are familiar with the target system, or simply trial and error.

After the first stage of fuzzy modeling, we obtain a rule base that can more or less describe the behavior of the target system by means of linguistic terms. The meaning of these linguistic terms is determined in the second stage, the identification of "deep structure", which determines the MFs of each linguistic term (and the coefficients of each rule's output in the case that a Sugeno model is used). Specifically, the identification of deep structure includes the following tasks:

1. Choose an appropriate family of parameterized MFs.
2. Interview human experts familiar with the target systems to determine the parameters of the MFs used in the rule base.
3. Refine the parameters of the MFs using regression and optimization techniques.

Task 1 and 2 assume the availability of human experts, while task 3 assumes the availability of a desired input-output data set. When a fuzzy inference system is used as a controller for a given plant, then the objective in task 3 should be changed to that of searching for parameters that will generate the best performance of the plant.

2.5 Summary

In this chapter, we have presented the main ideas underlying type-1 fuzzy logic and we have only started to point out the many possible applications of this powerful computational theory. We have discussed in some detail fuzzy set theory, fuzzy reasoning and fuzzy inference systems. At the end, we also gave some remarks about fuzzy modeling. In the following chapters, we will show how fuzzy logic techniques (in some cases, in conjunction with other methodologies) can be applied to solve real world complex problems.

3 Type-2 Fuzzy Logic

We introduce in this chapter a new area in fuzzy logic, which is called type-2 fuzzy logic. Basically, a type-2 fuzzy set is a set in which we also have uncertainty about the membership function. Of course, type-2 fuzzy systems consist of fuzzy if-then rules, which contain type-2 fuzzy sets. We can say that type-2 fuzzy logic is a generalization of conventional fuzzy logic (type-1) in the sense that uncertainty is not only limited to the linguistic variables but also is present in the definition of the membership functions.

Fuzzy Logic Systems are comprised of rules. Quite often, the knowledge that is used to build these rules is uncertain. Such uncertainty leads to rules whose antecedents or consequents are uncertain, which translates into uncertain antecedent or consequent membership functions (Karnik & Mendel 1998). Type-1 fuzzy systems (like the ones seen in the previous chapter), whose membership functions are type-1 fuzzy sets, are unable to directly handle such uncertainties. We describe in this chapter, type-2 fuzzy systems, in which the antecedent or consequent membership functions are type-2 fuzzy sets. Such sets are fuzzy sets whose membership grades themselves are type-1 fuzzy sets; they are very useful in circumstances where it is difficult to determine an exact membership function for a fuzzy set.

The original fuzzy logic, founded by Lotfi Zadeh, has been around for more than 30 years, and yet it is unable to handle uncertainties (Mendel, 2001). That the original fuzzy logic (type-1 fuzzy logic) cannot do this sounds paradoxical because the word "fuzzy" has the connotation of uncertainty. The expanded fuzzy logic (type-2 fuzzy logic) is able to handle uncertainties because it can model and minimize their effects.

In what follows, we shall first introduce the basic concepts of type-2 fuzzy sets, and type-2 fuzzy reasoning. Then we will introduce and compare the different types of fuzzy inference systems that have been employed in various applications. We will also consider briefly type-2 fuzzy logic systems and the comparison to type-1 fuzzy systems.

3.1 Type-2 Fuzzy Sets

The concept of a type-2 fuzzy set, was introduced by Zadeh (1975) as an extension of the concept of an ordinary fuzzy set (henceforth called a "type-1 fuzzy set"). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership grade

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 29–43, 2008.
springerlink.com © Springer-Verlag Berlin Heidelberg 2008

for each element of this set is a fuzzy set in [0,1], unlike a type-1 set where the membership grade is a crisp number in [0,1]. Such sets can be used in situations where there is uncertainty about the membership grades themselves, e.g., an uncertainty in the shape of the membership function or in some of its parameters. Consider the transition from ordinary sets to fuzzy sets. When we cannot determine the membership of an element in a set as 0 or 1, we use fuzzy sets of type-1. Similarly, when the situation is so fuzzy that we have trouble determining the membership grade even as a crisp number in [0,1], we use fuzzy sets of type-2.

This does not mean that we need to have extremely fuzzy situations to use type-2 fuzzy sets. There are many real-world problems where we cannot determine the exact form of the membership functions, e.g., in time series prediction because of noise in the data. Another way of viewing this is to consider type-1 fuzzy sets as a first order approximation to the uncertainty in the real-world. Then type-2 fuzzy sets can be considered as a second order approximation. Of course, it is possible to consider fuzzy sets of higher types but the complexity of the fuzzy system increases very rapidly. For this reason, we will only consider very briefly type-2 fuzzy sets. Lets consider some simple examples of type-2 fuzzy sets.

Example 3.1. Consider the case of a fuzzy set characterized by a Gaussian membership function with mean m and a standard deviation that can take values in $[\sigma_1,\sigma_2]$, i.e.,

$$\mu(x)=\exp\{-½[(x-m)/\sigma]^2\}; \qquad \sigma \in [\sigma_1,\sigma_2] \tag{3.1}$$

Corresponding to each value of σ, we will get a different membership curve (see Figure 3.1). So, the membership grade of any particular x (except x=m) can take any of a number of possible values depending upon the value of σ, i.e., the membership grade is not a crisp number, it is a fuzzy set. Figure 3.1 shows the domain of the fuzzy set associated with x=0.7; however, the membership function associated with this fuzzy set is not shown in the figure.

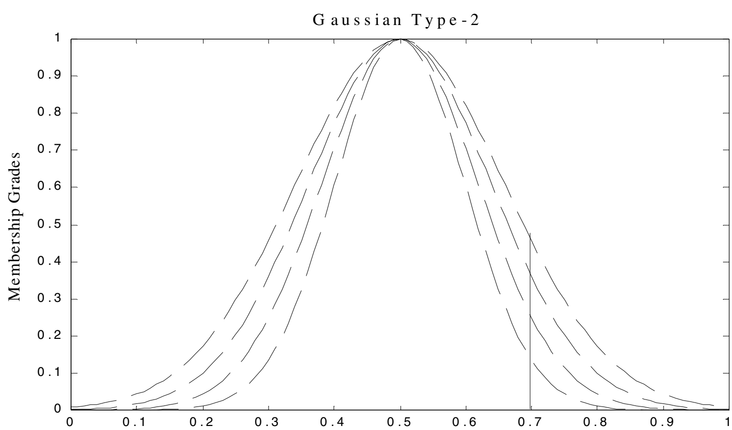


Fig. 3.1. A type-2 fuzzy set representing a type-1 fuzzy set with uncertain standard deviation

Example 3.2. Consider the case of a fuzzy set with a Gaussian membership function having a fixed standard deviation σ, but an uncertain mean, taking values in $[m_1, m_2]$, i.e.,

$$\mu(x)=\exp\{-½[(x-m)/\sigma]^2\}; \qquad m \in [m_1,m_2] \tag{3.2}$$

Again, μ(x) is a fuzzy set. Figure 3.2 shows an example of such a set.

Example 3.3. Consider a type-1 fuzzy set characterized by a Gaussian membership function (mean M and standard deviation σ_x), which gives one crisp membership m(x) for each input x ∈ X, where

$$m(x)=\exp\{-½\,[(x-M)/\sigma_x]^2\} \tag{3.3}$$

This is shown in Figure 3.3. Now, imagine that this membership of x is a fuzzy set. Let us call the domain elements of this set "primary memberships" of x (denoted by μ_1) and membership grades of these primary memberships "secondary memberships" of x [denoted by $\mu_2(x, \mu_1)$]. So, for a fixed x, we get a type-1 fuzzy set whose domain elements are primary memberships of x and whose corresponding membership grades are secondary memberships of x. If we assume that the secondary memberships follow a Gaussian with mean m(x) and standard deviation σ_m, as in Figure 3.3, we can describe the secondary membership function for each x as

$$\mu_2(x,\mu_1) = e - ½\,[(\mu_1 - m(x))/\,\sigma_m]^2 \tag{3.4}$$

where $\mu_1 \in [0,1]$ and m is as in equation (3.3).

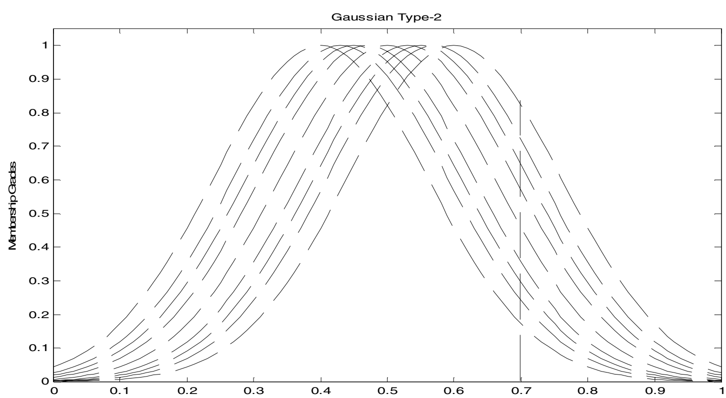


Fig. 3.2. A type-2 fuzzy set representing a type-1 fuzzy set with uncertain mean. The mean is uncertain in the interval [0.4, 0.6].

We can formally define these two kinds of type-2 sets as follows.

Definition 3.1. Gaussian type-2

A Gaussian type-2 fuzzy set is one in which the membership grade of every domain point is a Gaussian type-1 set contained in [0,1].

Example 3.3 shows an example of a Gaussian type-2 fuzzy set. Another way of viewing type-2 membership functions is in a three-dimensional fashion, in which we

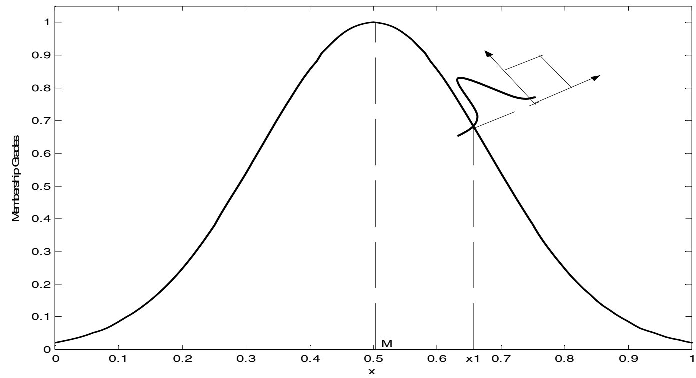


Fig. 3.3. A type-2 fuzzy set in which the membership grade of every domain point is a Gaussian type-1 set

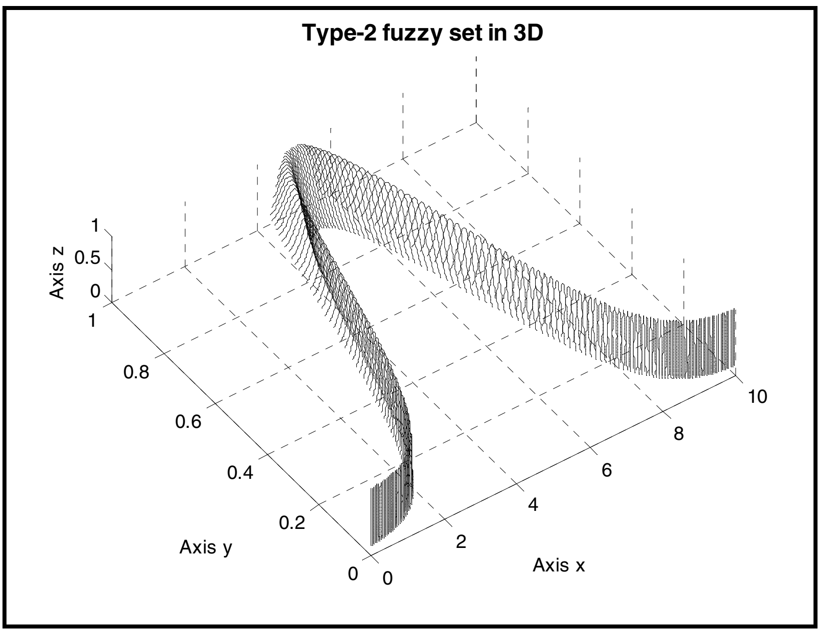


Fig. 3.4. Three-dimensional view of a type-2 membership function

can better appreciate the idea of type-2 fuzziness. In Figure 3.4 we have a three-dimensional view of a type-2 Gaussian membership function.

Definition 3.2. Interval type-2

An interval type-2 fuzzy set is one in which the membership grade of every domain point is a crisp set whose domain is some interval contained in [0,1].

Example 3.1 shows an example of an interval type-2 fuzzy set.

We will give some useful definitions on type-2 fuzzy sets in the following lines.

Definition 3.3. Footprint of uncertainty
Uncertainty in the primary memberships of a type-2 fuzzy set, Ã, consists of a bounded region that we call the "footprint of uncertainty" (FOU). Mathematically, it is the union of all primary membership functions (Mendel 2001).

We show as an illustration in Figure 3.5 the footprint of uncertainty for a type-2 Gaussian membership function. This footprint of uncertainty can be obtained by projecting in two dimensions the three-dimensional view of the type-2 Gaussian membership function.

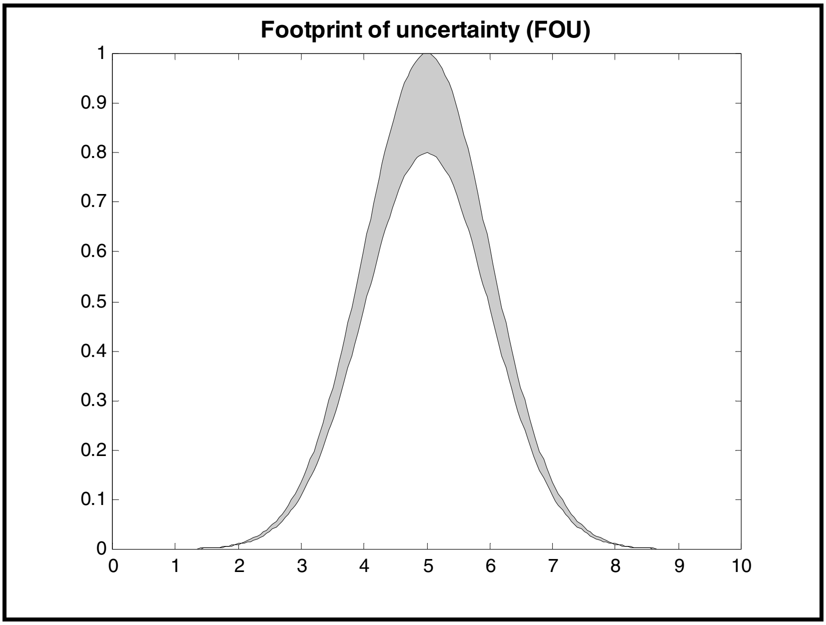


Fig. 3.5. Footprint of uncertainty of a sample type-2 Gaussian membership function

Definition 3.4. Upper and lower membership functions
An "upper membership function" and a "lower membership functions" are two type-1 membership functions that are bounds for the FOU of a type-2 fuzzy set Ã. The upper membership function is associated with the upper bound of FOU(Ã). The lower membership function is associated with the lower bound of FOU(Ã).

We illustrate the concept of upper and lower membership functions as well as the footprint of uncertainty in the following example.

Example 3.4. Gaussian primary MF with uncertain standard deviation
For the Gaussian primary membership function with uncertain standard deviation (Figure 3.1), the upper membership function is

$$upper(FOU(\tilde{A})) = N(m, \sigma_2; x) \tag{3.5}$$

And the lower membership function is

$$lower(FOU(\tilde{A})) = N(m, \sigma_1; x). \tag{3.6}$$

We will describe the operations and properties of type-2 fuzzy sets in the following section.

3.2 Operations of Type-2 Fuzzy Sets

In this section we describe the set theoretic operations of type-2 fuzzy sets. We are interested in the case of type-2 fuzzy sets, $\tilde{A}_i$ (i = 1,…,r), whose secondary membership functions are type-1 fuzzy sets. To compute the union, intersection, and complement of type-2 fuzzy sets, we need to extend the binary operations of minimum (or product) and maximum, and the unary operation of negation, from crisp numbers to type-1 fuzzy sets, because at each x, $\mu_{\tilde{A}i}$ (x, u) is a function (unlike the type-1 case, where $\mu_{\tilde{A}i}$ (x) is a crisp number). The tool for computing the union, intersection, and complement of type-2 fuzzy sets is Zadeh's extension principle (Zadeh, 1975).

Consider two type-2 fuzzy sets $\tilde{A}_1$ and $\tilde{A}_2$, i.e.,

$$\tilde{A}_1 = \int_x \mu_{\tilde{A}1}(x)/ x \tag{3.7}$$

and

$$\tilde{A}_2 = \int_x \mu_{\tilde{A}2}(x)/ x \tag{3.8}$$

In this section, we focus our attention on set theoretic operations for such general type-2 fuzzy sets.

Definition 3.5. Union of type-2 fuzzy sets

The union of $\tilde{A}_1$ and $\tilde{A}_2$ is another type-2 fuzzy set, just as the union of type-1 fuzzy sets A_1 and A_2 is another type-1 fuzzy set. More formally, we have the following expression

$$\tilde{A}_1 \cup \tilde{A}_2 = \int_{x\in X} \mu_{\tilde{A}1\cup\tilde{A}2}(x)/ x \tag{3.9}$$

We can explain Equation (3.9) by the "join" operation (Mendel, 2001). Basically, the join between two secondary membership functions must be performed between every possible pair of primary memberships. If more than one combination of pairs gives the same point, then in the join we keep the one with maximum membership grade. We will consider a simple example to illustrate the union operation. In Figure 3.6 we plot two type-2 Gaussian membership functions, and the union is shown in Figure 3.7.

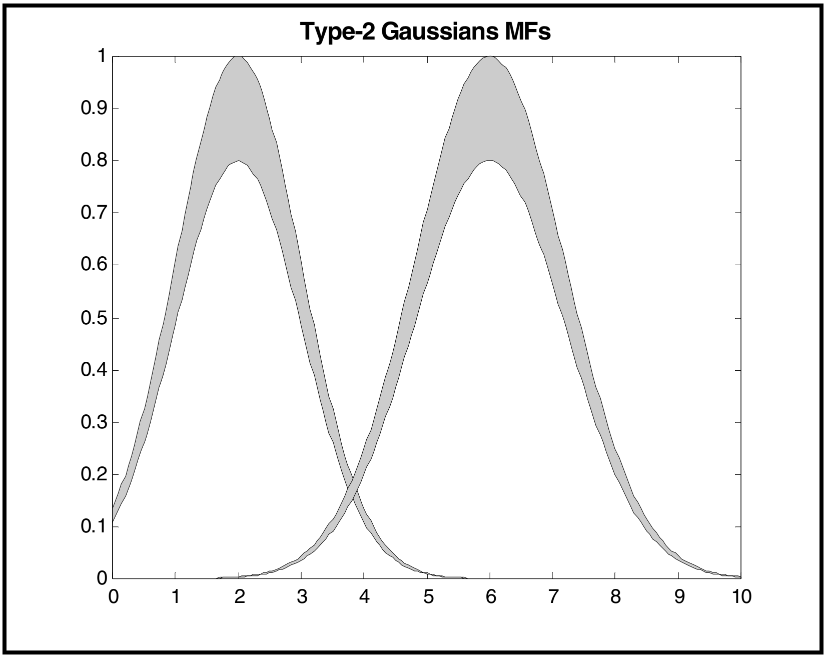


Fig. 3.6. Two sample type-2 Gaussian membership functions

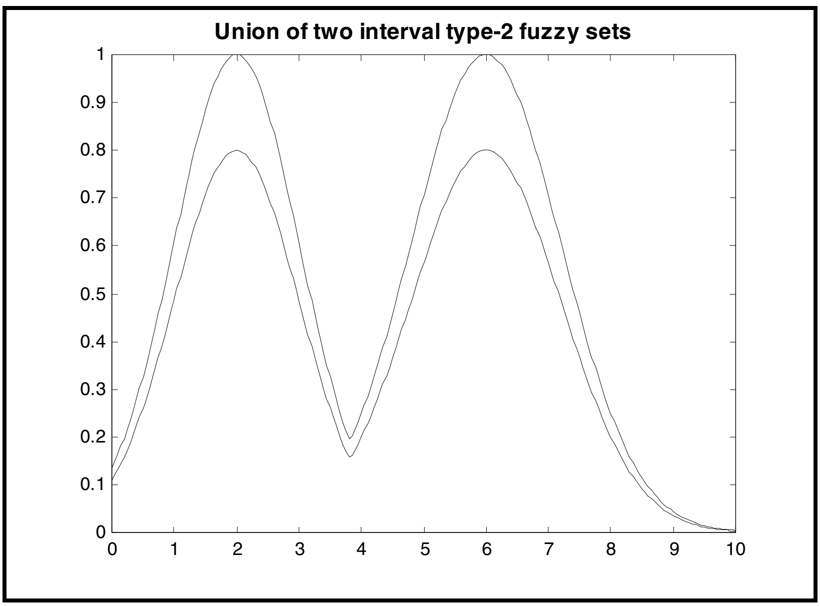


Fig. 3.7. Union of the two Gaussian membership functions

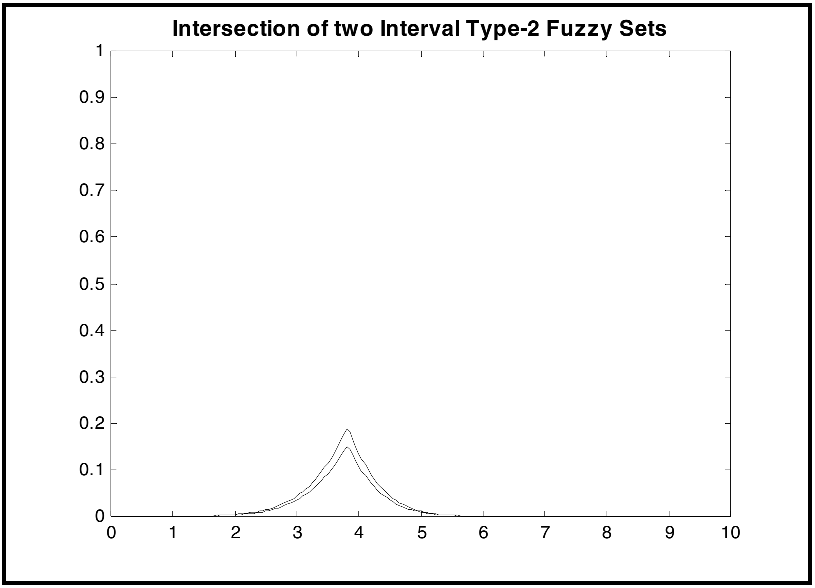


Fig. 3.8. Intersection of two type-2 Gaussian membership functions

Definition 3.6. Intersection of type-2 fuzzy sets
The intersection of $\tilde{A}_1$ and $\tilde{A}_2$ is another type-2 fuzzy set, just as the intersection of type-1 fuzzy sets A_1 and A_2 is another type-1 fuzzy set. More formally, we have the following expression

$$\tilde{A}_1 \cap \tilde{A}_2 = \int_{x \in X} \mu_{\tilde{A}1 \cap \tilde{A}2}(x)/\, x \quad (3.10)$$

We illustrate the intersection of two type-2 Gaussian membership functions in Figure 3.8

We can explain Equation (3.10) by the "meet" operation (Mendel, 2001). Basically, the meet between two secondary membership functions must be performed between every possible pair of primary memberships. If more than one combination of pairs gives the same point, then in the meet we keep the one with maximum membership grade.

Definition 3.7. Complement of a type-2 fuzzy set
The complement of set $\tilde{A}$ is another type-2 fuzzy set, just as the complement of type-1 fuzzy set A is another type-1 fuzzy set. More formally we have

$$\tilde{A}' = \int_x \mu_{\tilde{A}'1}(x)/\, x \quad (3.11)$$

where the prime denotes complement in the above equation. In this equation $\mu_{\tilde{A}'1}$ is a secondary membership function, i.e., at each value of x $\mu_{\tilde{A}'1}$ is a function (unlike the type-1 case where, at each value of x, $\mu_{\tilde{A}'1}$ is a point value).

Example 3.5. Type-2 fuzzy set operations
In this example we illustrate the union, intersection and complement operations for two type-2 fuzzy sets $\tilde{A}_1$ and $\tilde{A}_2$, and for a particular element x for which the

secondary membership functions in these two sets are $\mu_{\tilde{A}1}(x) = 0.5/0.1 + 0.8/0.2$ and $\mu_{\tilde{A}2}(x) = 0.4/0.5 + 0.9/0.9$. Using in the operations the minimum t-norm and the maximum t-conorm, we have the following results:

$$\begin{aligned}\mu_{\tilde{A}1\cup\tilde{A}2}(x) = \mu_{\tilde{A}1}(x) \cup \mu_{\tilde{A}2}(x) &= (0.5/0.1 + 0.8/0.2) \cup (0.4/0.5 + 0.9/0.9)\\ &= (0.5 \wedge 0.4)/(0.1 \vee 0.5) + (0.5 \wedge 0.9)/(0.1 \vee 0.9) + (0.8 \wedge 0.4)/(0.2 \vee 0.5) + (0.8 \wedge 0.9)/(0.2 \vee 0.9)\\ &= 0.4/0.5 + 0.5/0.9 + 0.4/0.5 + 0.8/0.9\\ &= \max\{0.4, 0.4\}/0.5 + \max\{0.5, 0.8\}/0.9\\ &= 0.4/0.5 + 0.8/0.9\end{aligned}$$

$$\begin{aligned}\mu_{\tilde{A}1\cap\tilde{A}2}(x) = \mu_{\tilde{A}1}(x) \cap \mu_{\tilde{A}2}(x) &= (0.5/0.1 + 0.8/0.2) \cap (0.4/0.5 + 0.9/0.9)\\ &= (0.5 \wedge 0.4)/(0.1 \wedge 0.5) + (0.5 \wedge 0.9)/(0.1 \wedge 0.9) + (0.8 \wedge 0.4)/(0.2 \wedge 0.5) + (0.8 \wedge 0.9)/(0.2 \wedge 0.9)\\ &= 0.4/0.1 + 0.5/0.1 + 0.4/0.2 + 0.8/0.2\\ &= \max\{0.4, 0.5\}/0.1 + \max\{0.4, 0.8\}/0.2\\ &= 0.5/0.1 + 0.8/0.2\end{aligned}$$

$$\mu_{\tilde{A}'1}(x) = 0.5/(1-0.1) + 0.8/(1-0.2) = 0.5/0.9 + 0.8/0.8.$$

3.3 Type-2 Fuzzy Systems

The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general, will not change for any type-n (Karnik & Mendel 1998). A higher-type number just indicates a higher "degree of fuzziness". Since a higher type changes the nature of the membership functions, the operations that depend on the membership functions change; however, the basic principles of fuzzy logic are independent of the nature of membership functions and hence, do not change. Rules of inference like Generalized Modus Ponens or Generalized Modus Tollens continue to apply.

The structure of the type-2 fuzzy rules is the same as for the type-1 case because the distinction between type-2 and type-1 is associated with the nature of the membership functions. Hence, the only difference is that now some or all the sets involved in the rules are of type-2. In a type-1 fuzzy system, where the output sets are type-1 fuzzy sets, we perform defuzzification in order to get a number, which is in some sense a crisp (type-0) representative of the combined output sets. In the type-2 case, the output sets are type-2; so we have to use extended versions of type-1 defuzzification methods. Since type-1 defuzzification gives a crisp number at the output of the fuzzy system, the extended defuzzification operation in the type-2 case gives a type-1 fuzzy set at the output. Since this operation takes us from the type-2 output sets of the fuzzy system to a type-1 set, we can call this operation "type reduction" and call the type-1 fuzzy set so obtained a "type-reduced set". The type-reduced fuzzy set may then be defuzzified to obtain a single crisp number; however, in many applications, the type-reduced set may be more important than a single crisp number.

Type-2 sets can be used to convey the uncertainties in membership functions of type-1 fuzzy sets, due to the dependence of the membership functions on available linguistic and numerical information. Linguistic information (e.g. rules from experts), in general, does not give any information about the shapes of the membership functions. When membership functions are determined or tuned based on numerical data, the uncertainty in the numerical data, e.g., noise, translates into uncertainty in the membership functions. In all such cases, any available information about the linguistic/numerical uncertainty can be incorporated in the type-2 framework. However, even with all of the advantages that fuzzy type-2 systems have, the literature on the applications of type-2 sets is scarce. Some examples are Yager (1980) for decision making, and Wagenknecht & Hartmann (1988) for solving fuzzy relational equations. We think that more applications of type-2 fuzzy systems will come in the near future as the area matures and the theoretical results become more understandable for the general public in the fuzzy arena.

3.3.1 Singleton Type-2 Fuzzy Logic Systems

This section discusses the structure of a singleton type-2 Fuzzy Logic Systems (FLS), which is a system that accounts for uncertainties about the antecedents or consequents in rules, but does not explicitly account for input measurement uncertainties. More complicated (but, more versatile) non-singleton type-2 FLSs, which account for both types of uncertainties, are discussed later.

The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general will 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 membership functions, the operations that depend on the membership functions change, however, the basic principles of fuzzy logic are independent of the nature of membership functions and hence do not change. Rules of inference, like Generalized Modus Ponens, continue to apply.

A general type-2 FLS is shown in Figure 3.9. As discussed before a type-2 FLS is very similar to type-1 FLS, the major structural difference being that the defuzzifier block of a type-1 FLS is replaced by the output processing block in type-2 FLS. That block consists of type-reduction followed by defuzzification.

During our development of a type-2 FLS, we assume that all the antecedent and consequent sets in rules are type-2, however, this need not necessarily be the case in practice. All results remain valid as long as long as just one set is type-2. This means that a FLS is type-2 as long as any one of its antecedent or consequent sets is type-2.

In the type-1 case, we generally have fuzzy if-then rules of the form

$$R^l: \text{IF } x_1 \text{ is } A^l_1 \text{ and } \ldots\ x_p \text{ is } A^l_p, \text{ THEN } y \text{ is } Y^l \qquad l = 1,\ldots, M \tag{3.12}$$

As mentioned earlier, the distinction between type-1 and type-2 is associated with the nature of the membership functions, which is not important when forming the rules. 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.

Consider a type-2 FLS having r inputs $x_1 \in X_1, \ldots, x_r \in X_r$ and one output $y \in Y$. As in the type-1 case, we can assume that there are M rules; but, in the type-2 case the *l*th rule has the form

$$R^l: \text{IF } x_1 \text{ is } \tilde{A}^l_1 \text{ and } \ldots\ x_p \text{ is } \tilde{A}^l_p, \text{ THEN } y \text{ is } \hat{Y}^l \qquad l = 1,\ldots, M \tag{3.13}$$

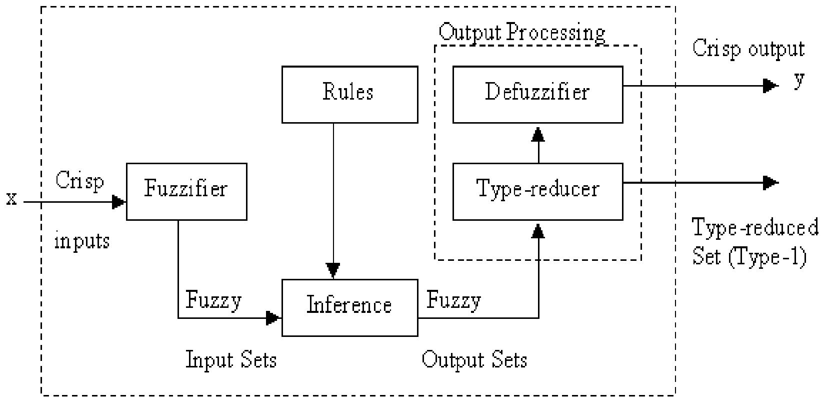


Fig. 3.9. Type-2 Fuzzy Logic System

This rule represents a type-2 fuzzy relation between the input space $X_1 \times \ldots \times X_r$, and the output space, Y, of the type-2 fuzzy system.

In a type-1 FLS the inference engine combines rules and gives a mapping from input type-1 fuzzy sets to output type-1 fuzzy sets. Multiple antecedents in rules are combined by the t-norm. The membership grades in the input sets are combined with those in the output sets using composition. Multiple rules may be combined using the t-conorm or during defuzzification by weighted summation. In the type-2 case the inference process is very similar. The inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. To do this one needs to compute unions and intersections of type-2 fuzzy sets, as well as compositions of type-2 relations.

In the type-2 fuzzy system (Figure 3.9), as in the type-1 fuzzy system, crisp inputs are first fuzzified into fuzzy input sets that then activate the inference block, which in the present case is associated with type-2 fuzzy sets. In this section, we describe singleton fuzzification and the effect of such fuzzification on the inference engine. The "fuzzifier" maps a crisp point $\mathbf{x} = (x_1,\ldots, x_r)^T \in X_1 \times X_2 \ldots \times X_r \equiv \mathbf{X}$ into a type-2 fuzzy set $\tilde{A}_x$ in $\mathbf{X}$.

The type-2 output of the inference engine shown in Figure 3.9 must be processed next by the output processor, the first operation of which is type-reduction. Type-reduction methods include (Mendel, 2001): centroid, center-of-sums, height, modified height, and center-of-sets. Lets assume that we perform centroid type-reduction. Then each element of the type-reduced set is the centroid of some embedded type-1 set for the output type-2 set of the FLS. Each of these embedded sets can be thought of as an output set of an associated type-1 FLS, and, correspondingly, the type-2 FLS can be viewed of as a collection of many different type-1 FLSs. Each type-1 FLS is embedded in the type-2 FLS; hence, the type-reduced set is a collection of the outputs of all of the embedded type-1 FLSs (see Figure 3.10). The type-reduced set lets us represent the output of the type-2 FLS as a fuzzy set rather than as a crisp number, which is something that cannot be done with a type-1 fuzzy system.

Referring to Figure 3.10, when the antecedent and consequent membership functions of the type-2 FLS have continuous domains, the number of embedded sets is uncountable. Figure 3.10 shows a situation in which we have assumed that the membership functions have discrete (or discretized) domains. The memberships in the type-reduced set, $\mu_Y(y_i)$, represent the level of uncertainty associated with each embedded type-1 FLS. A crisp output can be obtained by aggregating the outputs of all embedded type-1 FLSs by, e.g., finding the centroid of the type-reduced set.

If all of the type-2 uncertainties were to disappear, the secondary membership functions for all antecedents and consequents would each collapse to a single point, which shows that the type-2 FLS collapses to a type-1 FLS.

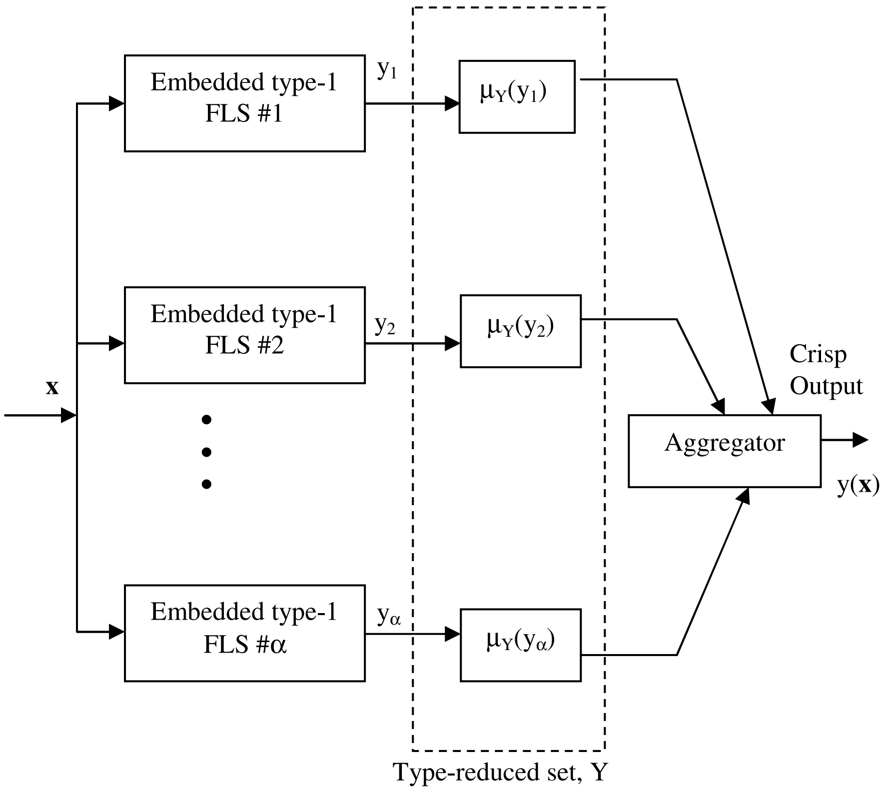


Fig. 3.10. A type-2 FLS viewed as a collection of embedded type-1 FLSs

If we think of a type-2 FLS as a “perturbation” of a type-1 FLS, due to uncertainties in their membership functions, then the type-reduced set of the type-2 FLS can be thought of as representing the uncertainty in the crisp output due to the perturbation. Some measure of the spread of the type-reduced set may then be taken to indicate the possible variation in the crisp output due to the perturbation. This is analogous to using confidence intervals in a stochastic-uncertainty situation.

We defuzzify the type-reduced set to get a crisp output from the type-2 FLS. The most natural way to do this seems to be finding the centroid of the type-reduced set. Finding the centroid is equivalent to finding the weighted average of the outputs of all the type-1 FLSs that are embedded in the type-2 FLS, where the weights correspond to the memberships in the type-reduced set (see Figure 3.10). If the type-reduced set Y for an input **x** is discretized or is discrete and consists of α points, then the expression for its centroid is

$$y(\mathbf{x}) = [\Sigma^{\alpha}_{k=1} \, y_k \, \mu_Y(y_k)\,] / \, [\Sigma^{\alpha}_{k=1} \, \mu_Y(y_k)\,] \tag{3.14}$$

If α is large then data storage may be a problem for the computation of Equation (3.14). This equation can, however, be evaluated using parallel processing, in this case data storage will not be problem. Currently, however, most researchers still depend on software for simulations and cannot make use of parallel processing. We can, however, use a recursive method to vastly reduce the memory required for storing the data that are needed to compute the defuzzification output. From Equation (3.14), we can calculate

$$A(i) = A(i\text{-}1) + y_i \, \mu_Y(y_i) \qquad A(0) = 0 \tag{3.15}$$

and

$$B(i) = B(i\text{-}1) + y_i \, \mu_Y(y_i) \qquad B(0) = 0 \tag{3.16}$$

for i =1,…, α. With these formulas we just need to store A and B during each iteration.

From our previous discussions about the five elements that comprise the Figure 3.9 type-2 FLS, we see that there are many possibilities to choose from, even more than for a type-1 FLS. To begin, we must decide on the kind of defuzzification (singleton or non-singleton). We must also choose a FOU for each type-2 membership function, decide on the functional forms for both the primary and secondary membership functions, and choose the parameters of the membership functions (fixed a-priori or tuned during a training procedure). Then we need to choose the composition (max-min, max-product), implication (minimum, product), type-reduction method (centroid, center-of-sums, height, modified height, center-of-sets), and defuzzifier. Clearly, there is an even greater richness among type-2 FLSs than there is among type-1 FLSs. In other words, there are more design degrees of freedom associated with a type-2 FLS than with a type-1 FLS; hence, a type-2 FLS has the potential to outperform a type-1 FLS because of the extra degrees of freedom.

3.3.2 Non-singleton Fuzzy Logic Systems

A non-singleton FLS is one whose inputs are modeled as fuzzy numbers. A type-2 FLS whose inputs are modeled as type-1 fuzzy numbers is referred to as “type-1 non-singleton type-2 FLS”. This kind of a fuzzy system not only accounts for uncertainties about either the antecedents or consequents in rules, but also accounts for input measurement uncertainties.

A type-1 non-singleton type-2 FLS is described by the same diagram as in singleton type-2 FLS, see Figure 3.9. The rules of a type-1 non-singleton type-2 FLS are the same as those for the singleton type-2 FLS. What are different is the fuzzifier, which treats the inputs as type-1 fuzzy sets, and the effect of this on the inference block. The output of the inference block will again be a type-2 fuzzy set; so, the type-reducers and defuzzifier that we described for a singleton type-2 FLS apply as well to a type-1 non-singleton type-2 FLS.

We can also have a situation in which the input are modeled as type-2 fuzzy numbers. This situation can occur, e.g., in time series forecasting when the additive measurement noise is non-stationary. A type-2 FLS whose inputs are modeled as type-2 fuzzy numbers is referred to as “type-2 non-singleton type-2 FLS”.

A type-2 non-singleton type-2 FLS is described by the same diagram as in singleton type-2 FLS, see Figure 3.9. The rules of a type-2 non-singleton type-2 FLS are the same as those for a type-1 non-singleton type-2 FLS, which are the same as those for a singleton type-2 FLS. What is different is the fuzzifier, which treats the inputs as type-2 fuzzy sets, and the effect of this on the inference block. The output of the inference block will again be a type-2 fuzzy set; so, the type-reducers and defuzzifier that we described for a type-1 non-singleton type-2 FLS apply as well to a type-2 non-singleton type-2 FLS.

3.3.3 Sugeno Type-2 Fuzzy Systems

All of our previous FLSs were of the Mamdani type, even though we did not refer to them as such. In this section, we will need to distinguish between the two kinds of FLSs, we refer to our previous FLSs as “Mamdani” FLSs. Both kinds of FLS are characterized by if-then rules and have the same antecedent structures. They differ in the structures of their consequents. The consequent of a Mamdani rule is a fuzzy set, while the consequent of a Sugeno rule is a function.

A type-1 Sugeno FLS was proposed by Takagi and Sugeno (1985), and Sugeno and Kang (1988), in an effort to develop a systematic approach to generating fuzzy rules from a given input-output data set. We will consider in this section the extension of first-order type-1 Sugeno FLS to its type-2 counterpart, with emphasis on interval sets.

Consider a type-2 Sugeno FLS having r inputs $x_1 \in X_1, \ldots, x_r \in X_r$ and one output $y \in Y$. A type-2 Sugeno FLS is also described by fuzzy if-then rules that represent input-output relations of a system. In a general first-order type-2 Sugeno model with a rule base of M rules, each having r antecedents, the *i*th rule can be expressed as

$$R^1: \text{IF } x_1 \text{ is } \tilde{A}^1_1 \text{ and } \ldots\ x_p \text{ is } \tilde{A}^1_p, \text{ THEN } Y^i = C^i_0 + C^i_1\, x_1 + \ldots + C^i_r\, x_r \qquad (3.17)$$

where $i = 1,\ldots, M$; C^i_j $(j = 1,\ldots, r)$ are consequent type-1 fuzzy sets; Y^i, the output of the *i*th rule, is also a type-1 fuzzy set (because it is a linear combination of type-1 fuzzy sets); and $\tilde{A}^i_k$ $(k = 1, \ldots, r)$ are type-2 antecedent fuzzy sets. These rules let us simultaneously account for uncertainty about antecedent membership functions and consequent parameter values. For a type-2 Sugeno FLS there is no need for type-reduction, just as there is no need for defuzzification in a type-1 Sugeno FLS.

3.4 Summary

In this chapter, we have presented the main ideas underlying type-2 fuzzy logic and we have only started to point out the many possible applications of this powerful computational theory. We have discussed in some detail type-2 fuzzy set theory, fuzzy reasoning and fuzzy inference systems. At the end, we also gave some remarks about type-2 fuzzy modeling with the Mamdani and Sugeno approaches. In the following chapters, we will show how type-2 fuzzy logic (in some cases, in conjunction with other methodologies) can be applied to solve real world complex problems. This chapter will serve as a basis for the new hybrid intelligent methods, for modeling, simulation, and pattern recognition that will be described later this book.

4 A Method for Type-2 Fuzzy Inference in Control Applications

A novel method of type 2 fuzzy logic inference is presented in this chapter. The method is highly efficient regarding computational time and implementation effort. Type-2 input membership functions are optimized using the Human Evolutionary Model (HEM) considering as the objective function the Integral of Squared Error at the controllers output. Statistical tests were achieved considering how the error at the controller's output is diminished in presence of uncertainty, demonstrating that the proposed method outperforms an optimized traditional type-2 fuzzy controller for the same test conditions.

4.1 Introduction

In engineering as well as in the scientific field is of growing interest to use type-2 fuzzy logic controller (FLC). It is a well documented fact that type-2 FLC had demonstrated in several fields their usefulness to handle uncertainty which is an inherent characteristic of real systems. Because uncertainty and real systems are inseparable characteristics the research of novel methods to handle incomplete or not too reliable information is of great interest (Mendel, 2001). Recently, we have seen the use of type-2 fuzzy sets in Fuzzy Logic Systems (FLS) in different areas of application. From those including fuzzy logic systems, neural networks and genetic algorithms, to some papers with emphasis on the implementation of type-2 FLS; in others, it is explained how type-2 fuzzy sets let us model and minimize the effects of uncertainties in rule-base FLS (Mendel and John, 2002). Also, a paper that provides mathematical formulas and computational flowcharts for computing the derivatives that are needed to implement steepest-descent parameter tuning algorithms for type-2 fuzzy logic systems (Mendel, 2004). Some research works are devoted to solve real world applications in different areas, for example in signal processing, type-2 fuzzy logic is applied in prediction of the Mackey-Glass chaotic time-series with uniform noise presence (Karnik and Mendel, 1999). In medicine, an expert system was developed for solving the problem of Umbilical Acid-Base (UAB) assessment (Ozen and Garibaldi, 2003). In industry, type-2 fuzzy logic and neural networks was used in the control of

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 45–52, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

non-linear dynamic plants (Melin and Castillo, 2004); also we can find interesting studies in the field of mobile robots (Hagras, 2004).
Although, the use of a type-2 FLC can be considered as a viable option to handle uncertainty, also it is well known all the deficiencies and requirements that the use of this technology implies.

In this work we are presenting a method whose goal is to simplify the implementation of a type-2 FLC without any loss of reliability in the results. In fact, this novel method reduces some of the stressful difficult to implement the traditional type-2 FLC.

The organization of this work is as follows: In section 2 is explained step by step how to implement this proposal and the method used to optimize the traditional as well as the proposed type-2 FLC. Section 3 is devoted to explain the kind and classification of experiments that were achieved, also in this section are given the experimental results. In section 4 is performed a discussion about the obtained results. Finally, in section 5 we have the conclusions.

4.2 Proposed Method to Implement Type-2 FLC

It is proposed to use two type-1 fuzzy systems (FS) to emulate a type-2 FS. The membership functions (MF), fuzzification process, fuzzy inference and defuzzification are type-1. The MFs are organized in such a way that they will be able to emulate the footprint of uncertainty (FOU) in a type-2 FS. To obtain the best parametric values for the MF the proposed method uses the optimized MFs, and we used the Human Evolutionary Model (HEM) to achieve the optimization.

To validate the proposed method, we made several comparative experiments using type-1 fuzzy traditional systems, as well as type-2 interval FS in accordance to those worked by (Mendel, 2001). The tests were achieved in the experimental base shown in Figure 4.1 which is a closed loop control system. The control goal is to make a tracking of the input signal r, which is applied to the systems summing junction. Note that we are using an adaptive fuzzy controller that needs to be optimized. In the feedback, with the aim of proving the proposal, we are considering two situations. One is to directly connect the system output to one summing junction. The second is to introduce noise to simulate uncertainty in the feedback data. At the summing junction output we have the error signal, which is applied to the input of the fuzzy controller, from the error signal we are obtaining a derivative signal; i.e., the change of error vs. time, which also is applied to the controllers input.

In general, the proposed solution to substitute the Mendel's type-2 FS consists in using the average of two type-1 FS, and to achieve this is necessary to follow the next steps:

1. To substitute each type-2 MF with two type-1 MFs. For doing this, the FOU of each MF is substituted with two type-1 MFs. In Fig. 4.2, the error signal (input fuzzy variable) e consists of three linguistic variables, they have been substituted as was explained obtaining the fuzzy sets that are shown in Fig. 4.3 where each fuzzy set is a type-1 MF. The first type-1 FLC (FLC1) is constructed using the upper MFs, and the second one (FLC2) with the lower MFs.

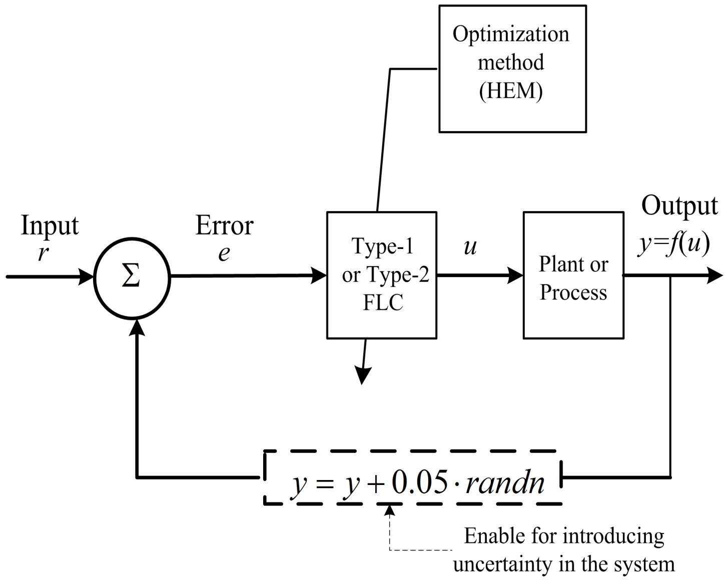


Fig. 4.1. Block diagram of the system used to test the proposal solution

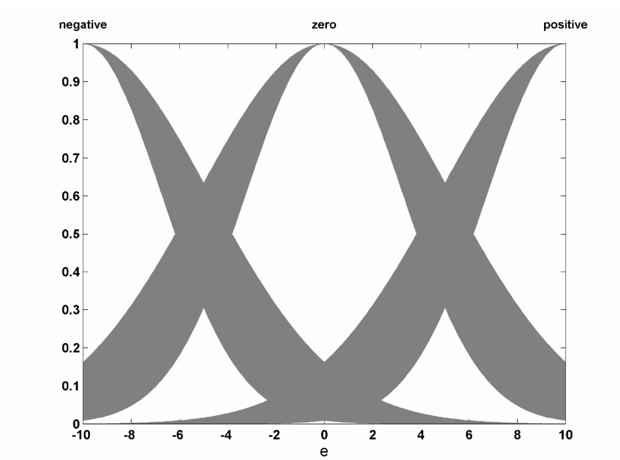


Fig. 4.2. Type-2 MF for the error input

2. To substitute the type-2 inference system, it is necessary to obtain the inference of each type-1 system in the traditional way. 1
3. To substitute the type reduction and defuzzification stages of a type-2 FS, it is necessary to obtain the defuzzification of each system as is traditionally done, and average them.

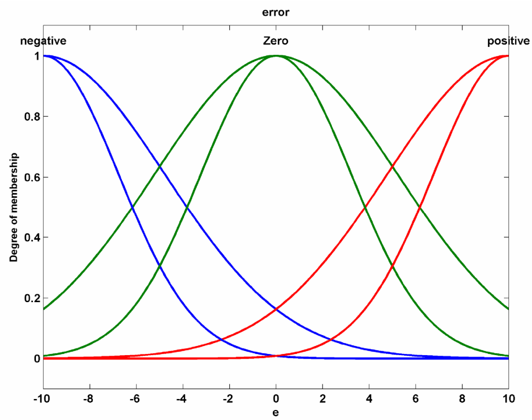


Fig 4.3. Substitution of the type-2 MFs of the error input using type-1 MFs

Performance Criteria

For evaluating the transient closed-loop response of a computer control system we can use the same criteria that normally are used for adjusting constants in PID (Proportional Integral Derivative) controllers. These are (Sepulveda et al., 2007):

Integral of Square Error (ISE).

$$\text{ISE} = \int_0^{\infty} [e(t)]^2 \, dt \tag{4.1}$$

Integral of the Absolute value of the Error (IAE).

$$\text{IAE} = \int_0^{\infty} | e(t) | \, dt \tag{4.2}$$

Integral of the Time multiplied by the Absolute value of the Error (ITAE).

$$\text{ITAE} = \int_0^{\infty} t \, | e(t) | \, dt \tag{4.3}$$

The selection of the criteria depends on the type of response desired, the errors will contribute different for each criterion, so we have that large errors will increase the value of ISE more heavily than to IAE. ISE will favor responses with smaller overshoot for load changes, but ISE will give longer settling time. In ITAE, time appears as a factor, and therefore, ITAE will penalize heavily errors that occur late in time, but virtually ignores errors that occur early in time.

4.3 Experiments

The experiments were divided in two classes:

1. The first class was to find, under different ranges for the FOU, the optimal values for the parameters of the interval type-2 MFs of the type-2 FLC of the non-linear control plant.
2. On the second class of experiments; it was realized the same as in the first class, but considering the average of the two type-1 FLC.

Class 1. Experiments with type-2 FLC

It is a fact that type-2 FLCs offer better conditions to handle uncertainty, so the purpose of the experiments of class 1, were to find the optimal parameters of the interval type-2 MFs to control the plant in a better way.

It was used a novel evolutive algorithm; Human Evolutionary Model (Montiel et al., 2005), to find those optimal values and to analize the influence of the FOU, we realized several tests for different ranges of it, beginning with the thinner and finally with the broader one. Once the optimal values were found, it was tested the behavior of the type-2 FLC, for different noise levels, from 8 db to 30 db.

Class 2. Experiments with average of two FLCs

To control the plant, we used the proposal solution of using the average of two type-1 FLC to simulate a type-2 FLC. For these experiments, it was considered that one type-1 FLC manage and fixed the upper MFs, and the other the low MFs. Here, in the same way as in experiments of class 1, from the optimal values found for the MFs, it was tested the behavior of the average of two type-1 FLC, for different noise levels, from 8 db to 30 db.

4.4 Results

The HEM was the optimization method that we used (Montiel et al., 2005). The initial setting for each range of the FOU for this evolutionary method were:

Initial population of individuals =20
Low bound of individuals=10
Upper bound of individuals=100
Number of variables=6 (Standard deviation of each of the MFs of the inputs).
Number of generations=60

The search process was repeated 30 times, always looking for the optimal parameter values to obtain the lowest ISE value.

Class 1

In figures 4.4 and 4.5 it can be seen the optimized MFs that obtained the best results in the control of the plant.

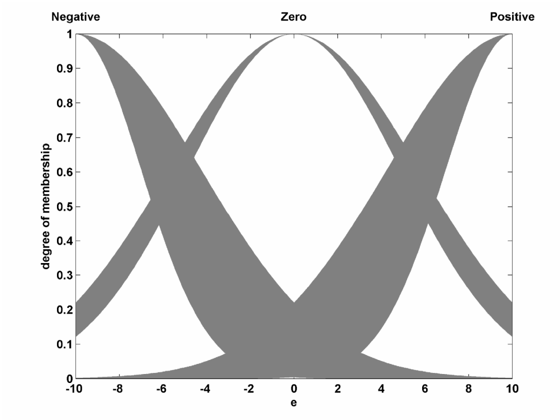


Fig. 4.4. Optimized MFs of the input error *e* of the type-2 FLC, for a 2.74 to 5.75 range of the FOU

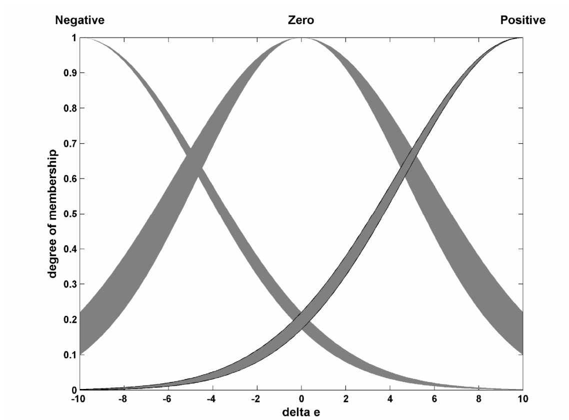


Fig. 4.5. Optimized MFs of the input delta-*e* of the type-2 FLC, for a 2.74 to 5.75 range of the FOU

Class 2

In figures 4.6 and 4.7, we can see the optimized MFs of the average of two type-1 FLCs, here as in Class 1, the best results were obtained in the broader range search.

Table 4.1, shows a comparison of the ISE values obtained for each FLC with its optimized MFs. As can be seen, with the proposal of two optimizedtype-1 FLCs, the ISE error is lower in all the search ranges.

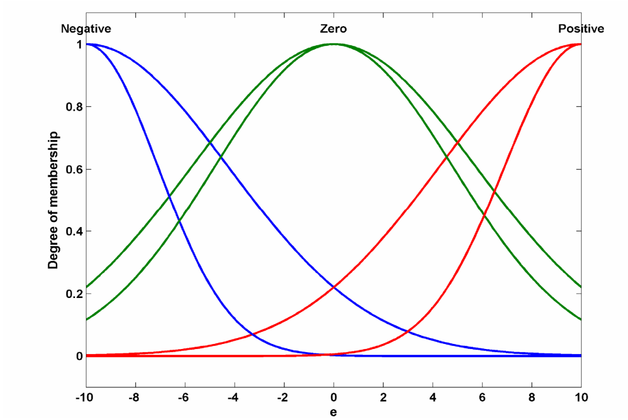


Fig. 4.6. Optimized MFs of the input error e of the average of two type-1 FLC, for a 2.74 to 5.75 range of the FOU

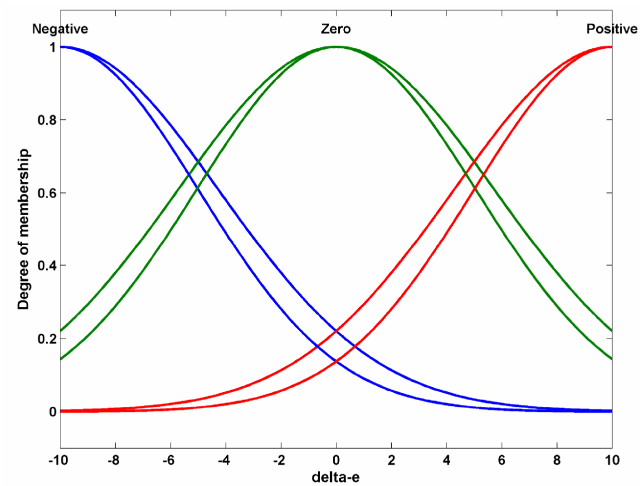


Fig. 4.7. Optimized MFs of the input delta-*e* of the two type-1 FLC, for a 2.74 to 5.75 range of the FOU

Table 4.1. Comparison values between Type-2 FLC and average of two type-1 FLCs

	TYPE-2 FLC		AVERAGE TYPE-1 FLCs	
Search range	Best ISE	AVERAGE ISE	Best ISE	**AVERAGE ISE**
3.74-4.75	4.761	4.9942	4.5619	**4.7701**
3.24-5.25	4.328	4.5060	4.2024	**4.4009**
2.74-5.75	**4.3014**	**4.4005**	**4.1950**	**4.346**

4.5 Summary

Based on the results of the experiments, we can conclude that the proposed method, that consists in using two optimized type-1 FLCs instead of a optimized traditional type-2 FLC, is a convenient and viable alternative because it offers advantages such as a highly efficient regarding computational time and implementation effort. The type-2 FLCs need to realize a complex task in each step of the process, specially in the type reduction case. With the proposed method it is easier to optimize the parameters of the MFs of a type-1 FLC than an interval type-2 FLC.

5 Design of Intelligent Systems with Interval Type-2 Fuzzy Logic

Uncertainty is an inherent part of intelligent systems used in real-world applications. The use of new methods for handling incomplete information is of fundamental importance. Type-1 fuzzy sets used in conventional fuzzy systems cannot fully handle the uncertainties present in intelligent systems. Type-2 fuzzy sets that are used in type-2 fuzzy systems can handle such uncertainties in a better way because they provide us with more parameters. This chapter deals with the design of intelligent systems using interval type-2 fuzzy logic for minimizing the effects of uncertainty produced by the instrumentation elements, environmental noise, etc. Experimental results include simulations of feedback control systems for non-linear plants using type-1 and type-2 fuzzy logic controllers; a comparative analysis of the systems' response is performed, with and without the presence of uncertainty.

5.1 Introduction

Uncertainty affects decision-making and appears in a number of different forms. The concept of information is fully connected with the concept of uncertainty. The most fundamental aspect of this connection is that the uncertainty involved in any problem-solving situation is a result of some information deficiency, which may be incomplete, imprecise, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way (Klir and Yuan, 1995). Uncertainty is an attribute of information (Zadeh, 2005). The general framework of fuzzy reasoning allows handling much of this uncertainty, fuzzy systems employ type-1 fuzzy sets, which represent uncertainty by numbers in the range [0, 1]. When something is uncertain, like a measurement, it is difficult to determine its exact value, and of course type-1 fuzzy sets make more sense than using sets (Zadeh, 1975). However, it is not reasonable to use an accurate membership function for something uncertain, so in this case what we need is another type of fuzzy sets, those which are able to handle these uncertainties, the so called type-2 fuzzy sets (Mizumoto and Tanaka, 1976) (Mendel, 2001). So, the amount of uncertainty in a system can be reduced by using type-2 fuzzy logic because it offers better capabilities to handle linguistic uncertainties by modeling vagueness and unreliability of information (Liang and Mendel, 2000).

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 53–76, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

Recently, we have seen the use of type-2 fuzzy sets in Fuzzy Logic Systems (FLS) in different areas of application (Lee et al., 2003). A novel approach for realizing the vision of ambient intelligence in ubiquitous computing environments (UCEs), is based on embedding intelligent agents that use type-2 fuzzy systems which are able to handle the different sources of uncertainty and imprecision in UCEs to give a good response (Doctor et al., 2005). There are also papers with emphasis on the implementation of type-2 FLS (Karnik and Mendel, 1999) and in others, it is explained how type-2 fuzzy sets let us model and minimize the effects of uncertainties in rule-base FLS (Wu and Mendel, 2001). There is also a paper that provides mathematical formulas and computational flowcharts for computing the derivatives that are needed to implement steepest-descent parameter tuning algorithms for type-2 fuzzy logic systems (Mendel, 2004). Some research works are devoted to solve real world applications in different areas, for example in signal processing, type-2 fuzzy logic is applied in prediction of the Mackey-Glass chaotic time-series with uniform noise presence (Mendel, 2000). In medicine, an expert system was developed for solving the problem of Umbilical Acid-Base (UAB) assessment (Ozen and Garibaldi, 2003). In industry, type-2 fuzzy logic and neural networks was used in the control of non-linear dynamic plants (Melin and Castillo, 2004); also we can find interesting studies in the field of mobile robots (Hagras, 2004).

In this chapter we deal with the application of interval type-2 fuzzy control to non-linear dynamic systems. It is a well known fact, that in the control of real systems, the instrumentation elements (instrumentation amplifier, sensors, digital to analog, analog to digital converters, etc.) introduce some sort of unpredictable values in the information that has been collected (Castillo and Melin, 2004). So, the controllers designed under idealized conditions tend to behave in an inappropriate manner (Castillo and Melin, 2001). Since, uncertainty is inherent in the design of controllers for real world applications, we are presenting how to deal with this problem using type-2 Fuzzy Logic Controller (FLC), to reduce the effects of imprecise information. We are supporting this statement with experimental results, qualitative observations, and quantitative measures of errors. For quantifying the errors, we utilized three widely used performance criteria, these are: Integral of Square Error (ISE), Integral of the Absolute value of the Error (IAE), and Integral of the Time multiplied by the Absolute value of the Error (ITAE) (Deshpande and Ash, 1988). We also consider the application of interval type-2 fuzzy logic to the problem of forecasting chaotic time series.

5.2 Fuzzy Logic Systems

In this section, a brief overview of type-1 and type-2 fuzzy systems is presented. This overview is considered as necessary to understand the basic concepts needed to understand the methods and algorithms presented later in the chapter.

5.2.1 Type-1 Fuzzy Logic Systems

In the 40's and 50's, many researchers proved that dynamic systems could be mathematically modeled using differential equations. In these works we have the foundations of the Control Theory, which in addition with the Transform Theory (Laplace's Theory), provided an extremely powerful means of analyzing and designing control systems (Mamdani, 1993). These theories were developed until the 70's, when the area was called Systems Theory to indicate its definitiveness.

Soft computing techniques have become an important research topic, which can be applied in the design of intelligent controllers (Jang et al., 1997). These techniques have tried to avoid the above-mentioned drawbacks, and they allow us to obtain efficient controllers, which utilize the human experience in a more natural form than the conventional mathematical approach (Zadeh, 1973). In the cases in which a mathematical representation of the controlled system is difficult to obtain, the process operator has the knowledge, the experience to express the relationships existing in the process behavior.

A FLS, described completely in terms of type-1 fuzzy sets is called a type-1 fuzzy logic system (type-1 FLS). It is composed by a knowledge base, which comprises the information given by the process operator in form of linguistic control rules. A fuzzification interface, that has the effect of transforming crisp data into fuzzy sets. An inference system, that uses the fuzzy sets in conjunction with the knowledge base to make inferences by means of a reasoning method. Finally, a defuzzification interface, which translates the fuzzy control action so obtained to a real control action using a defuzzification method (Mendel, 2001).

In this chapter, the implementation of the fuzzy controller in terms of type-1 fuzzy sets, has two input variables, which are the error $e(t)$, the difference between the reference signal and the output of the process, as well as the error variation $\Delta e(t)$,

$$e(t) = r(t) - y(t) \tag{5.1}$$

$$\Delta e(t) = e(t) - e(t-1) \tag{5.2}$$

so the control system can be represented as in Figure 5.1.

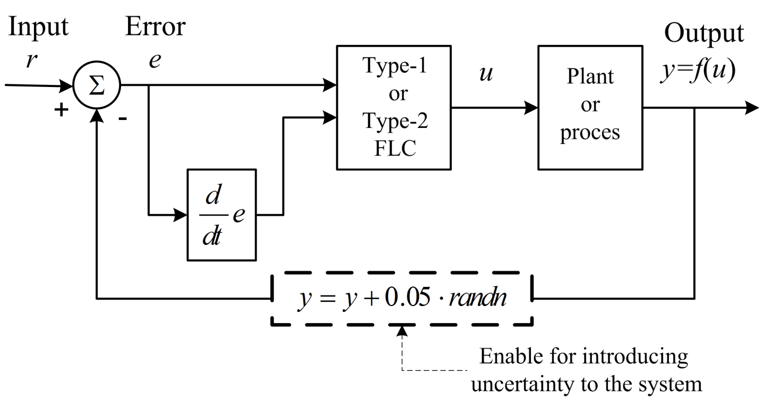


Fig. 5.1. System used for obtaining the experimental results for control

5.2.2 Type-2 Fuzzy Logic Systems

If for a type-1 membership function, as in Figure 5.2, we blur it to the left and to the right, as illustrated in Figure 5.3, then a type-2 membership function is obtained. In this case, for a specific value x', the membership function (u'), takes on different values, which are not all weighted the same, so we can assign an amplitude distribution to all of those points.

Doing this for all $x \in X$, we create a three-dimensional membership function –a type-2 membership function– that characterizes a type-2 fuzzy set (Mendel, 2001)

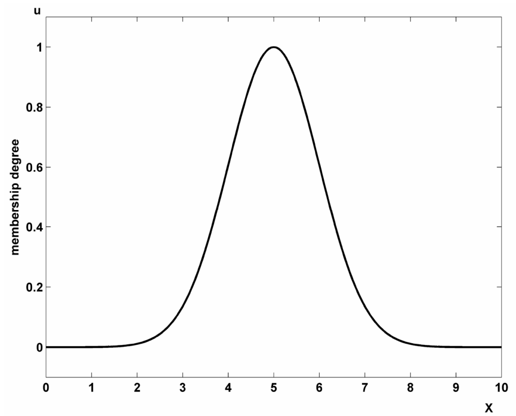


Fig. 5.2. Type-1 membership function

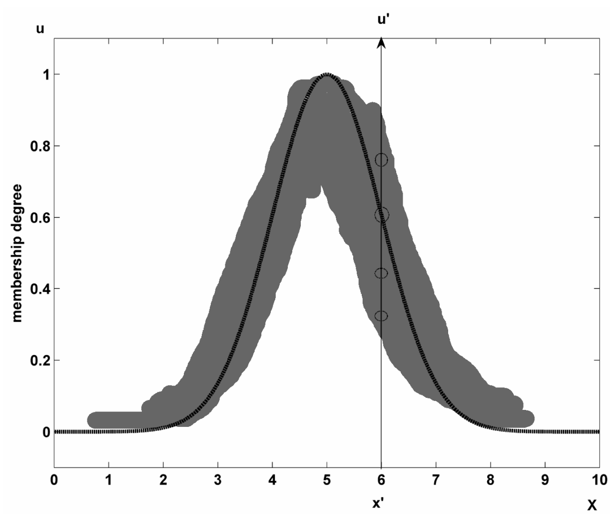


Fig. 5.3. Blurred type-1 membership function

(Mendel and Mouzouris, 1999). A type-2 fuzzy set $\tilde{A}$, is characterized by the membership function:

$$\tilde{A} = \{((x,u),\mu_{\tilde{A}}(x,u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \quad (5.3)$$

in which $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$. Another expression for $\tilde{A}$ is,

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \qquad J_x \subseteq [0,1] \quad (5.4)$$

Where $\int\int$ denotes the union over all admissible input variables *x* and *u*. For discrete universes of discourse $\int$ is replaced by $\sum$ (Mendel and John, 2002). In fact $J_x \subseteq [0,1]$ represents the primary membership of *x*, and $\mu_{\tilde{A}}(x,u)$ is a type-1 fuzzy set known as the secondary set. Hence, a type-2 membership grade can be any subset in [0,1], the primary membership, and corresponding to each primary membership, there is a secondary membership (which can also be in [0,1]) that defines the possibilities for the primary membership (Liang and Mendel, 2000). Uncertainty is represented by a region, which is called the footprint of uncertainty (FOU). When $\mu_{\tilde{A}}(x,u) = 1, \forall u \in J_x \subseteq [0,1]$ we have an interval type-2 membership function, as shown in Figure 5.4. The uniform shading for the FOU represents the entire interval type-2 fuzzy set and it can be described in terms of an upper membership function $\bar{\mu}_{\tilde{A}}(x)$ and a lower membership function $\underline{\mu}_{\tilde{A}}(x)$.

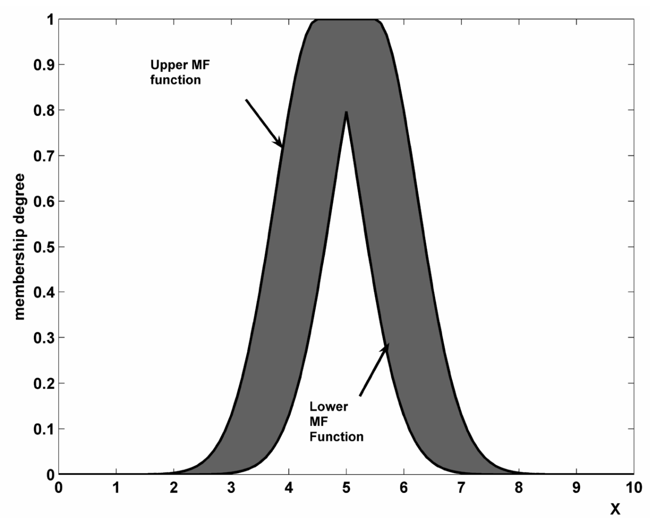


Fig. 5.4. Interval type-2 membership function

A FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain. On the other hand, type-2 FLSs, are very useful in circumstances where it is difficult to determine an exact membership function, and there are measurement uncertainties (Mendel, 2001).

It is known that type-2 fuzzy sets enable modeling and minimizing the effects of uncertainties in rule-based FLS. Unfortunately, type-2 fuzzy sets are more difficult to use and understand than type-1 fuzzy sets; hence, their use is not widespread yet. As a justification for the use of type-2 fuzzy sets, in (Sepulveda et al., 2007) are mentioned at least four sources of uncertainties not considered in type-1 FLSs:

1. The meanings of the words that are used in the antecedents and consequents of rules can be uncertain (words mean different things to different people).
2. Consequents may have histogram of values associated with them, especially when knowledge is extracted from a group of experts who do not all agree.
3. Measurements that activate a type-1 FLS may be noisy and therefore uncertain.
4. The data used to tune the parameters of a type-1 FLS may also be noisy.

All of these uncertainties translate into uncertainties about fuzzy set membership functions. Type-1 fuzzy sets are not able to directly model such uncertainties because their membership functions are totally crisp. On the other hand, type-2 fuzzy sets are able to model such uncertainties because their membership functions are themselves fuzzy. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity.

A type-2 FLS is again characterized by IF-THEN rules, but its antecedent or consequent sets are now of type-2. Type-2 FLSs, can be used when the circumstances are too uncertain to determine exact membership grades such as when the training data is corrupted by noise. Similar to a type-1 FLS, a type-2 FLS includes a fuzzifier, a rule

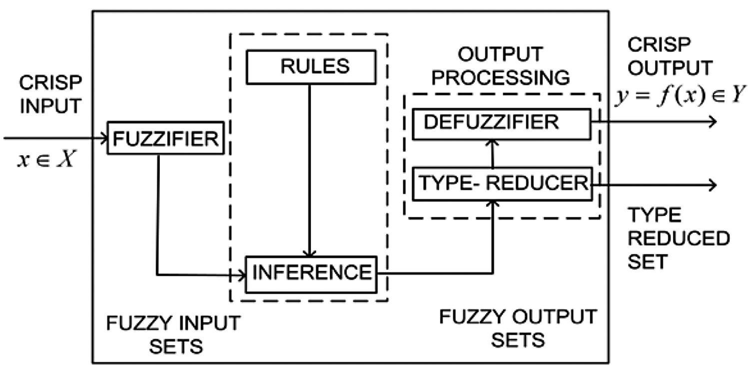


Fig. 5.5. Type-2 Fuzzy Logic System

base, fuzzy inference engine, and an output processor, as we can see in Fig. 5.5. The output processor includes type-reducer and defuzzifier; it generates a type-1 fuzzy set output (from the type-reducer) or a crisp number (from the defuzzifier) (Mendel, 2005). Now we will explain each of the blocks of Figure 5.5.

5.2.2.1 Fuzzifier

The fuzzifier maps a crisp point $\mathbf{x}=(x_1,\ldots,x_p)^T \in X_1 x X_2 x \ldots x X_p \equiv \mathbf{X}$ into a type-2 fuzzy set $\tilde{A}_x$ in **X** (Mendel, 2001), interval type-2 fuzzy sets in this case. We will use type-2 singleton fuzzifier, in a singleton fuzzification, the input fuzzy set has only a single point on nonzero membership. $\tilde{A}_x$ is a type-2 fuzzy singleton if $\mu_{\tilde{A}_x}(x) = 1/1$ for $\mathbf{x}=\mathbf{x'}$ and $\mu_{\tilde{A}_x}(x) = 1/0$ for all other $\mathbf{x}\neq\mathbf{x'}$[7].

5.2.2.2 Rules

The structure of rules in a type-1 FLS and a type-2 FLS is the same, but in the latter the antecedents and the consequents will be represented by type-2 fuzzy sets. So for a type-2 FLS with p inputs $x_1 \in X_1,\ldots,x_p \in X_p$ and one output $y \in Y$, Multiple Input Single Output (MISO), if we assume there are M rules, the lth rule in the type-2 FLS can be written as follows:

$$R^l\text{: IF } x_1 \text{ is } \tilde{F}_1^{\,l} \text{ and } \cdots \text{and } x_p \text{ is } \tilde{F}_p^{\,l} \text{ , THEN } y \text{ is } \tilde{G}^l \qquad l=1,\ldots,M \tag{5.5}$$

5.2.2.3 Inference

In the type-2 FLS, the inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. It is necessary to compute the join $\sqcup$, (unions) and the meet $\sqcap$ (intersections), as well as extended sup-star compositions (sup star compositions) of type-2 relations. If $\tilde{F}^l{}_1 \times\cdots\times \tilde{F}^l{}_p = \tilde{A}^l$, equation (5.5) can be re-written as

$$R^l : \tilde{F}^l{}_1 \times\cdots\times \tilde{F}^l{}_p \rightarrow \tilde{G}^l = \tilde{A}^l \rightarrow \tilde{G}^l \quad l=1,\ldots,M \tag{5.6}$$

R^l is described by the membership function $\mu_{R^l}(\mathbf{x}, y) = \mu_{R^l}(x_1,\ldots,x_p, y)$, where

$$\mu_{R^l}(\mathbf{x}, y) = \mu_{\tilde{A}^l \rightarrow \tilde{G}^l}(\mathbf{x}, y) \tag{5.7}$$

can be written as (Mendel, 2001):

$$\begin{aligned}\mu_{R^l}(\mathbf{x}, y) &= \mu_{\tilde{A}^l \rightarrow \tilde{G}^l}(\mathbf{x}, y) = \mu_{\tilde{F}_1^l}(x_1) \sqcap\cdots\sqcap \mu_{\tilde{F}_p{}^l}(x_p) \sqcap \mu_{\tilde{G}^l}(y) \\ &= [\sqcap_{i=1}^{p} \mu_{\tilde{F}^l{}_i}(x_i)] \sqcap \mu_{\tilde{G}^l}(y)\end{aligned} \tag{5.8}$$

In general, the p-dimensional input to R^l is given by the type-2 fuzzy set $\tilde{A}_x$ whose membership function is

$$\mu_{\tilde{A}_x}(\mathbf{x}) = \mu_{\tilde{x}_1}(x_1) \sqcap \cdots \sqcap \mu_{\tilde{x}p}(x_p) = \sqcap_{i=1}^{p} \mu_{\tilde{x}i}(x_i) \tag{5.9}$$

where $\tilde{X}_i (i = 1,..., p)$ are the labels of the fuzzy sets describing the inputs. Each rule R^l determines a type-2 fuzzy set $\tilde{B}^l = \tilde{A}_x \circ R^l$ such that:

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{A}_x \circ R^l} = \sqcup_{x \in \mathbf{X}} \left[\mu_{\tilde{A}_x}(\mathbf{x}) \sqcap \mu_{R^l}(\mathbf{x}, y) \right] \qquad y \in Y \;\; l = 1, \ldots, M \tag{5.10}$$

This equation is the input/output relation in Figure 5.5 between the type-2 fuzzy set that activates one rule in the inference engine and the type-2 fuzzy set at the output of that engine (Mendel, 2001).

In the FLS we used interval type-2 fuzzy sets and meet under product t-norm, so the result of the input and antecedent operations, which are contained in the firing set $\sqcap_{i=1}^{p} \mu_{\tilde{F}_{i_i}}(x'_i \equiv F^l(\mathbf{x}')$, is an interval type-1 set,

$$F^l(\mathbf{x}') = \left[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}') \right] \equiv \left[\underline{f}^l, \bar{f}^l \right] \tag{5.11}$$

Where

$$\underline{f}^l(\mathbf{x}') = \underline{\mu}_{\tilde{F}_1^l}(x_1') * \cdots * \underline{\mu}_{\tilde{F}_p^l}(x_p') \tag{5.12}$$

And

$$\bar{f}^l(\mathbf{x}') = \bar{\mu}_{\tilde{F}_1^l}(x_1') * \cdots * \bar{\mu}_{\tilde{F}_p^l}(x_p') \tag{5.13}$$

where * is the product operation.

5.2.2.4 Type Reducer

The type-reducer generates a type-1 fuzzy set output, which is then converted in a crisp output through the defuzzifier. This type-1 fuzzy set is also an interval set, for the case of our FLS we used center of sets (cos) type reduction, $Y_{\cos}$ which is expressed as (Mendel, 2001)

$$Y_{\cos}(\mathbf{x}) = [y_l, y_r] = \int_{y^1 \in [y_l^1, y_r^1]} \cdots \int_{y^M \in [y_l^M, y_r^M]} \int_{f^1 \in [\underline{f}^1, \bar{f}^1]} \cdots \int_{f^M \in [\underline{f}^M, \bar{f}^M]} 1 \Big/ \frac{\sum_{i=1}^{M} f^i y^i}{\sum_{i=1}^{M} f^i} \tag{5.14}$$

this interval set is determined by its two end points, y_l and y_r, which corresponds to the centroid of the type-2 interval consequent set $\tilde{G}^i$,

$$C_{\tilde{G}^i} = \int_{\theta_1 \in J_{y1}} \cdots \int_{\theta_N \in J_{yN}} 1 / \frac{\sum_{i=1}^{N} y_i \theta_i}{\sum_{i=1}^{N} \theta_i} = [y_l^{\ i}, y_r^{\ i}] \tag{5.15}$$

before the computation of $Y_{\cos}(\mathbf{x})$, we must evaluate equation (5.15), and its two end points, y_l and y_r. If the values of f_i and y_i that are associated with y_l are denoted f_l^i and y_l^i, respectively, and the values of f_i and y_i that are associated with y_r are denoted f_r^i and y_r^i, respectively, from equation (15.14), we have (Mendel, 2001)

$$y_l = \frac{\sum_{i=1}^{M} f_l^{\ i} y_l^{\ i}}{\sum_{i=1}^{M} f_l^{\ i}} \tag{5.16}$$

$$y_r = \frac{\sum_{i=1}^{M} f_r^{\ i} y_r^{\ i}}{\sum_{i=1}^{M} f_r^{\ i}} \tag{5.17}$$

5.2.2.5 Defuzzifier

From the type-reducer we obtain an interval set $Y_{\cos}$, to defuzzify it we use the average of y_l and y_r, so the defuzzified output of an interval singleton type-2 FLS is (Mendel, 2001)

$$y(\mathbf{x}) = \frac{y_l + y_r}{2} \tag{5.18}$$

In this chapter, we are simulating the fact that the instrumentation elements (instrumentation amplifier, sensors, digital to analog, analog to digital converters, etc.) are introducing some sort of unpredictable values in the collected information. In the case of the implementation of the type-2 FLC, we have the same characteristics as in type-1 FLC, but we used type-2 fuzzy sets as membership functions for the inputs and for the output.

5.2.3 Performance Criteria

For evaluating the transient closed-loop response of a computer control system we can use the same criteria that normally are used for adjusting constants in PID (Proportional Integral Derivative) controllers. These are (Sepulveda et al., 2007):

1. Integral of Square Error (ISE).

$$\text{ISE} = \int_0^{\infty} [e(t)]^2 \, dt \tag{5.19}$$

2 Integral of the Absolute value of the Error (IAE).

$$\text{IAE} = \int_0^{\infty} | e(t) | \, dt \tag{5.20}$$

3. Integral of the Time multiplied by the Absolute value of the Error (ITAE).

$$\text{ITAE} = \int_0^{\infty} t \, | e(t) | \, dt \tag{5.21}$$

The selection of the criteria depends on the type of response desired, the errors will contribute different for each criterion, so we have that large errors will increase the value of ISE more heavily than to IAE. ISE will favor responses with smaller overshoot for load changes, but ISE will give longer settling time. In ITAE, time appears as a factor, and therefore, ITAE will penalize heavily errors that occur late in time, but virtually ignores errors that occur early in time. Designing using ITAE will give us the shortest settling time, but it will produce the largest overshoot among the three criteria considered. Designing considering IAE will give us an intermediate result, in this case, the settling time will not be so large than using ISE nor so small than using ITAE, and the same applies for the overshoot response. The selection of a particular criterion is depending on the type of desired response.

5.3 Experimental Results for Intelligent Control

The experimental results are devoted to show comparisons in the system's response in a feedback controller when using a type-1 FLC or a type-2 FLC. A set of five experiments is described in this section. The first two experiments were performed in ideal conditions, i.e., without any kind of disturbance. In the last three experiments, Gaussian noise was added to the feedback loop with the purpose of simulating, in a global way, the effects of uncertainty from several sources.

Figure 5.1 shows the feedback control system that was used for obtaining the simulation results. The complete system was simulated in the Matlab programming language, and the controller was designed to follow the input as closely as possible. The plant is a non-linear system that is modeled using equation (5.22)

$$y(i) = 0.2 \cdot y(i-3) \cdot 0.07 y(i-2) + 0.9 \cdot y(i-1) + 0.05 u(i-1) + 0.5 \cdot u(i-2) \tag{5.22}$$

To illustrate the dynamics of this non-linear system, two different inputs are applied, first the input indicated by equation (5.23), which is shown in Figure 5.6, and whose system's response is in Figure 5.7.

$$u(i)=\begin{cases}0 & 1 \le i < 5 \\ .1 & 5 \le i < 10 \\ .5 & 10 \le i < 15 \\ 1 & 15 \le i < 20 \\ .5 & 20 \le i < 25 \\ 1 & 25 \le i < 30 \\ 0 & 30 \le i < 35 \\ 1.47 & 35 \le i < 40 \end{cases} \tag{5.23}$$

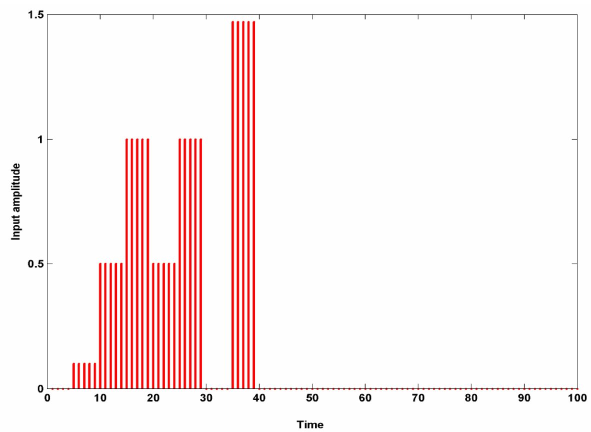


Fig. 5.6. Test sequence applied to the model of the plant given in equation (5.23)

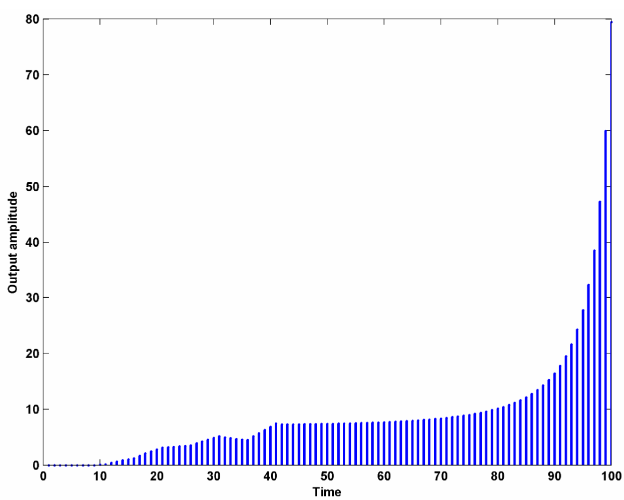


Fig. 5.7. System´s response for the inputs given in equation (5.23) which is illustrated in Fig. 5.6

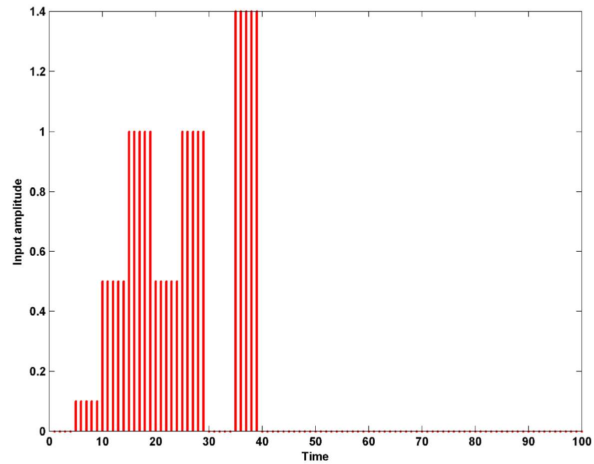


Fig. 5.8. A second input to the model for testing the plant response

Now, for a slightly different input given by equation (5.24), see Figure 5.8, we have the corresponding system´s response in Figure 5.9.

$$u(i)=\begin{cases} 0 & 1\le i<5 \\ .1 & 5\le i<10 \\ .5 & 10\le i<15 \\ 1 & 15\le i<20 \\ .5 & 20\le i<25 \\ 1 & 25\le i<30 \\ 0 & 30\le i<35 \\ 1.4 & 35\le i<40 \end{cases} \tag{5.24}$$

Going back to the control problem, this system given by equation (5.22) was used in Figure 5.1, under the name of plant or process, in this figure we can see that the controller's output is applied directly to the plant's input. Since we are interested in comparing the performance between type-1 and type-2 FLC systems, the controller was tested in two ways:

1. One is considering the system as ideal, that is, not introducing in the modules of the control system any source of uncertainty (experiments 1 and 2).
2. The other one is simulating the effects of uncertain modules (subsystems) response introducing some uncertainty (experiments 3, 4 and 5).

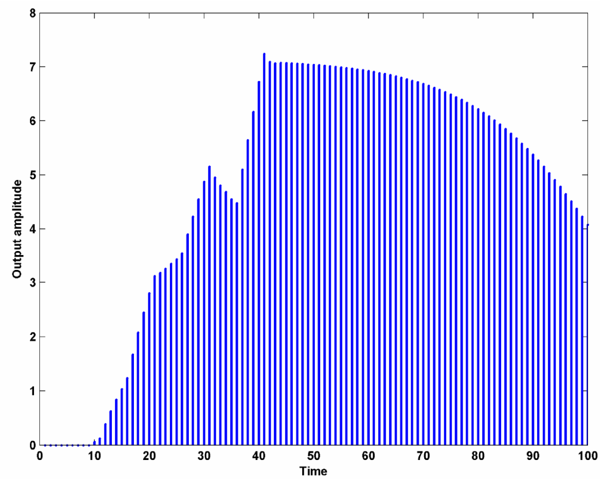


Fig. 5.9. Output of the plant when we applied the input given by equation (5.24) illustrated in Fig. 5.8

For both cases, as it is shown in Figure 5.1, the system's output is directly connected to the summing junction, but in the second case, the uncertainty was simulated introducing random noise with normal distribution (the dashed square in Figure 5.1). We added noise to the system's output $y(i)$ using the Matlab's function "randn", which generates random numbers with Gaussian distribution. The signal and the added noise in turn, were obtained with the programmer's expression (5.25), the result $y(i)$ was introduced to the summing junction of the controller system. Note that in expression (5.25) we are using the value 0.05, for experiments 3 and 4, but in the set of tests for experiment 5, we varied this value to obtain different SNR values.

$$y(i) = y(i) + 0.05 \cdot randn \tag{5.25}$$

The system was tested using as input, a unit step sequence free of noise, $r(i)$. For evaluating the system's response and comparing between type 1 and type 2 fuzzy controllers, the performance criteria ISE, IAE, and ITAE were used. In Table 5.3, we summarized the values obtained in an ideal system for each criterion considering 400 units of time. For calculating ITAE a sampling time of $T_s = 0.1$ sec. was considered.

For all experiments the reference input r is stable and noisy free. In experiments 3 and 4, although the reference appears clean, the feedback at the summing junction is noisy since noise for simulating the overall existing uncertainty in the system was introduced deliberately, in consequence, the controller's inputs $e(t)$ (error), and $\Delta e(t)$ contain uncertainty in the data.

In Experiment 5, we tested the systems, type-1 and type-2 FLCs, introducing different values of noise η, this was done by modifying the signal to noise ratio SNR (Proakis and Manolakis, 1996),

$$SNR = \frac{\sum |s|^2}{\sum |\eta|^2} = \frac{P_{signal}}{P_{noise}} \tag{5.26}$$

Because many signals have a very wide dynamic range (Ingle and Proakis, 2000), SNRs are usually expressed in terms of the logarithmic decibel scale, SNR(db),

$$SNR(db) = 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{5.27}$$

In Table 5.4, we show, for different values of SNR(db), the behavior of ISE, IAE, ITAE for type-1 and type-2 FLCs. In all the cases the results for type-2 FLC are better than type-1 FLC.

In the type-1 FLC, Gaussian membership functions (Gaussian MFs) for the inputs and for the output were used. A Gaussian MF is specified by two parameters {c,σ}:

$$\mu_A(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \tag{5.28}$$

c represents the MFs center and σ determines the MFs standard deviation.

For each of the inputs of the type-1 FLC, $e(t)$ and $\Delta e(t)$, three type-1 fuzzy Gaussian MFs were defined as: negative, zero, positive. The universe of discourse for these membership functions is in the range [-10 10]; their centers are -10, 0 and 10 respectively, and their standard deviations is 4.2466 as is illustrated in Figures 5.10 and 5.11.

For the output of the type-1 FLC, we have five type-1 fuzzy Gaussian MFs: NG, N, Z, P and PG. They are in the interval [-10 10], their centers are -10, -.5, 0, 5, and 10 respectively; and their standard deviation is 2.1233 as can be seen in Figure 5.12. Table 1 illustrates the characteristics of the MFs of the inputs and output of the type-1 FLC.

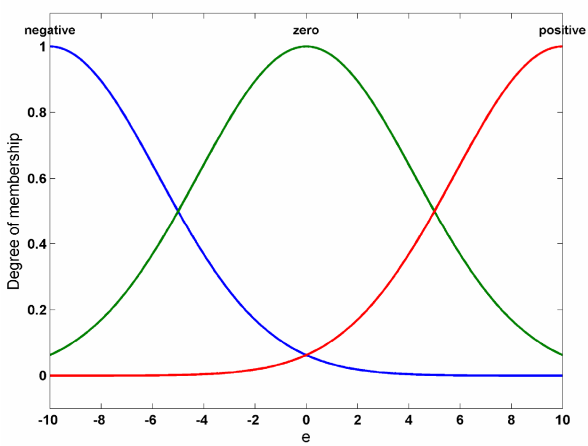


Fig. 5.10. Input e membership functions for the type-1 FLC

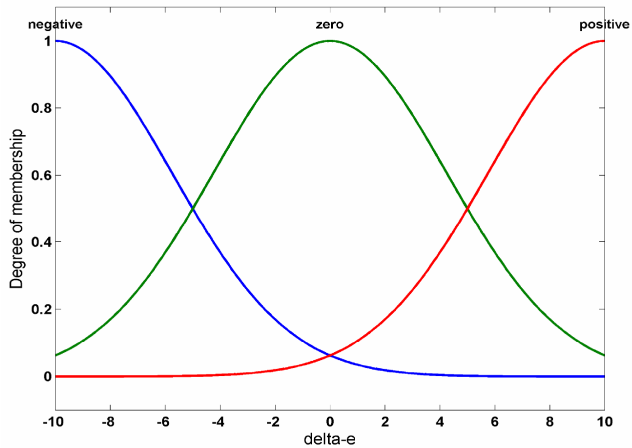


Fig. 5.11. Input Δe membership functions for the type-1 FLC

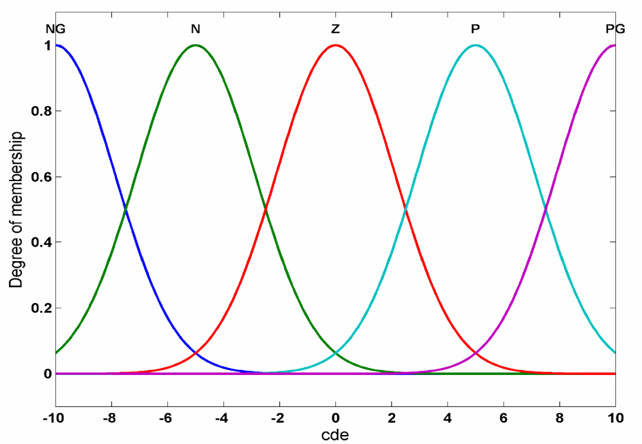


Fig. 5.12. Output cde membership functions for the type-1 FLC

In experiments 2, 4, and 5, for the type-2 FLC, as in type-1 FLC, we also selected Gaussian MFs for the inputs and for the output, but in this case we have interval type-2 Gaussian MFs with a fixed center, c, and an uncertain standard deviation, σ , i.e.,

$$\mu_A(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \tag{5.29}$$

In terms of the upper and lower membership functions, we have for $\mu_{\tilde{A}}(x)$,

$$\overline{\mu}_{\tilde{\mathrm{A}}}(\mathrm{x}) = \mathrm{N}(\mathrm{c}, \sigma_2; \mathrm{x}) \tag{5.30}$$

Table 5.1. Characteristics of the inputs and output of type-1 FLC

Variable	Term	Center c	Standard deviation σ
Input e	negative	-10	4.2466
	zero	0	4.2466
	positive	10	4.2466
Input Δe	Negative	-10	4.2466
	Zero	0	4.2466
	positive	10	4.2466
Output cde	NG	-10	2.1233
	N	-5	2.1233
	Z	0	2.1233
	P	5	2.1233
	PG	10	2.1233

and for the lower membership function $\underline{\mu}_{\tilde{A}}(x)$,

$$\underline{\mu}_{\tilde{A}}(x) = N(c, \sigma_1; x) \tag{5.31}$$

where $N(c, \sigma_2, x) \equiv e^{-\frac{1}{2}\left(\frac{x-c}{\sigma_2}\right)^2}$, and $N(c, \sigma_1, x) \equiv e^{-\frac{1}{2}\left(\frac{x-c}{\sigma_1}\right)^2}$, (Mendel, 2001). Hence, in the type-2 FLC, for each input we defined three interval type-2 fuzzy Gaussian MFs: negative, zero, positive in the interval [-10 10], as illustrated in Figures 5.13 and 5.14. For computing the output we have five interval type-2 fuzzy Gaussian

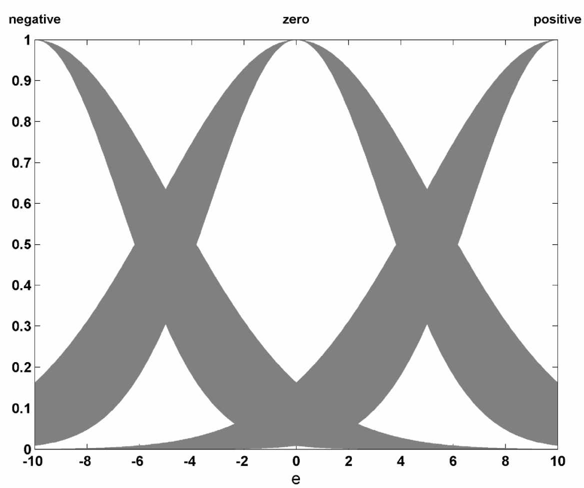


Fig. 5.13. Input e membership functions for the type-2 FLC

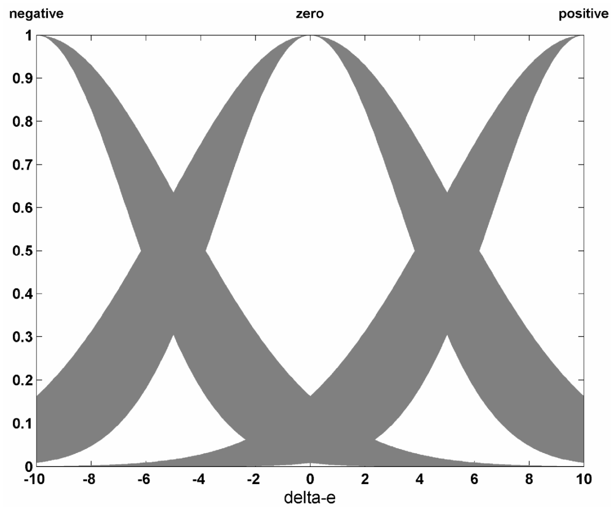


Fig. 5.14. Input Δe membership functions for the type-2 FLC

MFs, which are NG, N, Z, P and PG, in the interval [-10 10], as can be seen in Figures 5.15. Table 5.2 shows the characteristics of the inputs and output of the type-2 FLC.

For type-2 FLC we used, basically, the software for type-2 fuzzy logic developed by our research group. In all experiments, we have a dash-dot line for illustrating the system's response and behavior of type-1 FLC, in the same sense, a continuous line for type-2 FLC. The reference input *r* is shown with a dot line.

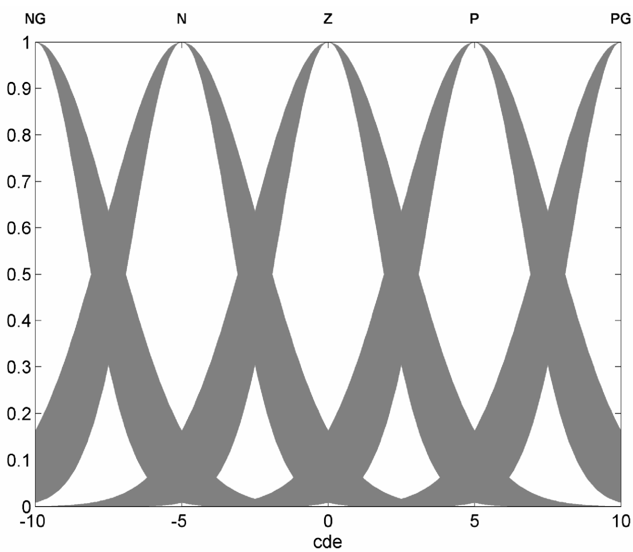


Fig. 5.15. Output cde membership functions for the type-2 FLC

Table 5.2. Characteristics of the inputs and output of type-2 FLC

Variable	Term	Center c	Standard deviation σ_1	Standard deviation σ_2
Input e	negative	-10	5.2466	3.2466
	zero	0	5.2466	3.2466
	positive	10	5.2466	3.2466
Input Δe	Negative	-10	5.2466	3.2466
	Zero	0	5.2466	3.2466
	positive	10	5.2466	3.2466
Output *cde*	NG	-10	2.6233	1.6233
	N	-5	2.6233	1.6233
	Z	0	2.6233	1.6233
	P	5	2.6233	1.6233
	PG	10	2.6233	1.6233

Experiment 1: Simulation of an ideal system with a type-1 FLC.
In this experiment, uncertainty data was not added to the system, and the system response is illustrated in Figure 5.16. Note that the settling time is of about 140 units of time; i.e., the system tends to stabilize with time and the output will follow accurately the input. In Table 5.3, we listed the obtained values of ISE, IAE, and ITAE for this

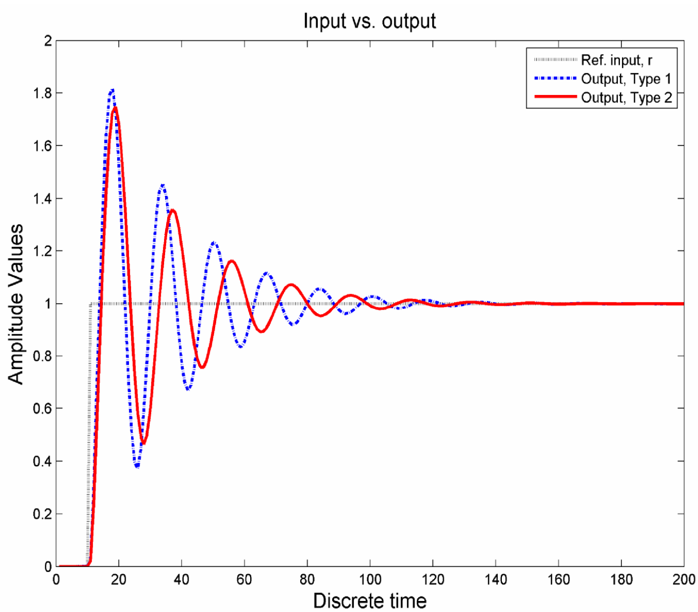


Fig. 5.16. This graphic shows the system's response to a unit step sequence. The input reference r is shown with pointed line, for the type-1 the systems' output $y(i)$ is shown with dash dot line; and for type-2, the system's output $y(i)$ with continuous line.

Table 5.3. Comparison of performance criteria for type-1 and type-2 fuzzy logic controllers for 20 db signal noise ratio. values obtained after 200 samples

Perform-ance Criteria	**Type-1 FLC**		**Type-2 FLC**	
	Ideal System	Syst. with uncertainty	Ideal System	Syst. with uncertainty
ISE	7.65	19.4	6.8	18.3
IAE	17.68	49.5	16.4	44.8
ITAE	62.46	444.2	56.39	402.9

experiment. In Figures 5.17, 5.18 and 5.19, the ISE, IAE, and ITAE behaviors of this experiment are shown.

Experiment 2: Simulation of an ideal system using the type-2 FLC.
Here, the same test conditions of Experiment 1 were used, but in this case, we implemented the controller's algorithm with type-2 fuzzy logic. The output sequence is illustrated in Figure 5.16, and the corresponding performance criteria are listed in Table 5.3, and we can observe that using a type-2 FLC we obtained the lower errors. By visual inspection, we can observe that the output system' response of the Experiment 1, and this one, are similar as it is shown in Figures 5.17, 5.18, and 5.19.

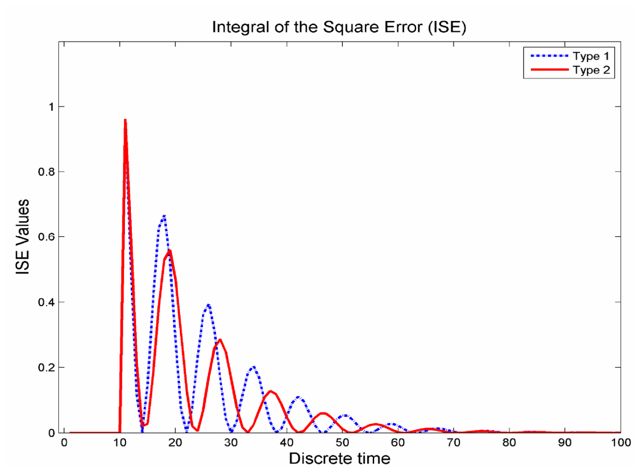


Fig. 5.17. In uncertainty absence, the ISE values are very similar for type-1 and type-2 FLCs

Experiment 3: System with uncertainty using a type-1 FLC.
In this case, equation (5.25) was used to simulate the effects of uncertainty introduced to the system by transducers, amplifiers, and any other element that in real world applications affects expected values. In this experiment the noise level was simulated in the range of 20 db of SNR ratio. Figure 5.20 shows the system's response output. In Figures 5.21, 5.22, and 5.23, the performance criteria ISE, IAE, ITAE are represented graphically.

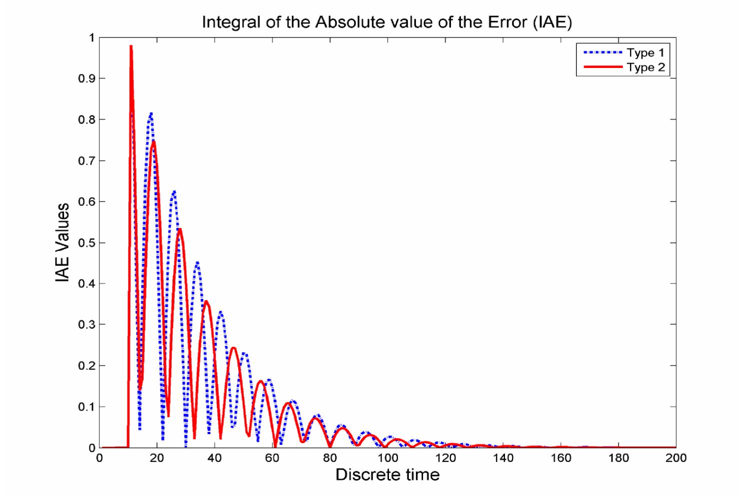


Fig. 5.18. In uncertainty absence, the IAE values obtained at the plant's output are very similar for type-1 and type-2 FLCs, here is more evident that a type-1 FLC works a little better than in Fig. 5.17

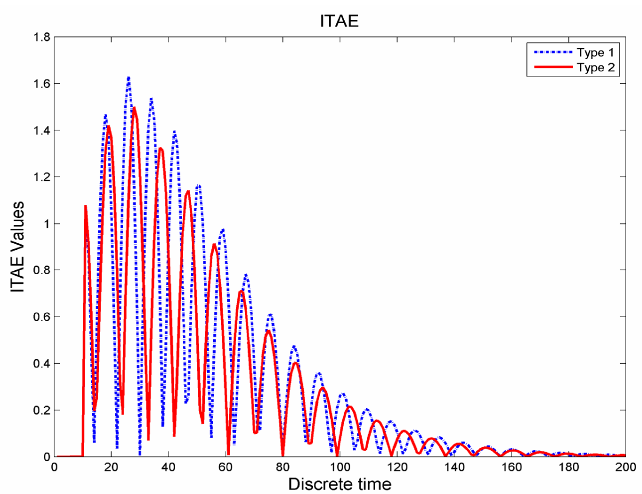


Fig. 5.19. In uncertainty absence, the ITAE values obtained at the plant's output are similar for type-1 and type-2 FLCs, in accordance with Figure 5.18, it is evident that a type-1 FLC works a little better

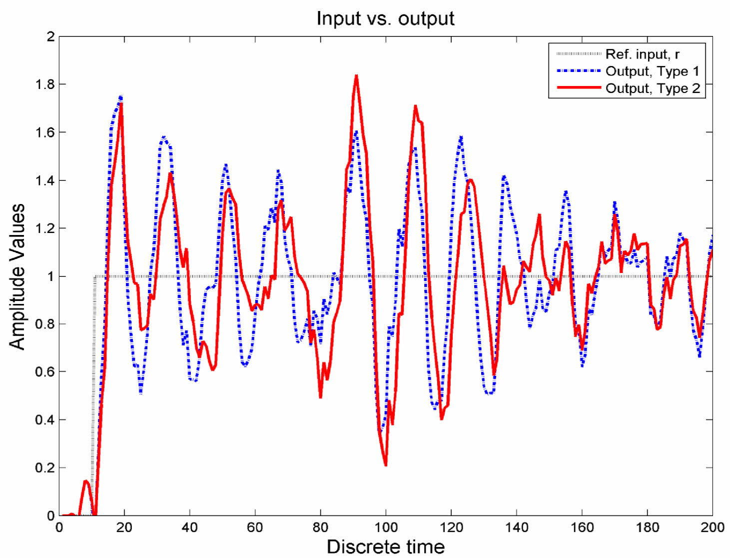


Fig. 5.20. This graphic was obtained with uncertainty presence; compare the system's outputs produced by type-1 and type-2 FLCs. Note that quite the opposite to Figure 5.16, a type-2 FLC works much better than a type-1 FLC when the system has uncertainty. The overshoot error is lower for a type-2 FLC.

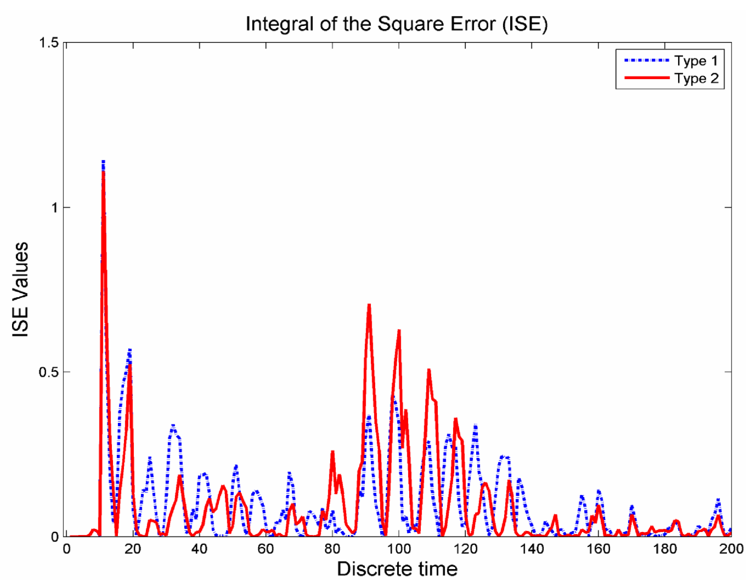


Fig. 5.21. We can see that a type-2 FLC produces lower overshoot errors, quantitatively the ISE overall error of using type-2 is 18.3 against 19.4 of the overall error produced by the type-1 FLC

Experiment 4: System with uncertainty using a type-2 FLC.
In this experiment, uncertainty was introduced in the system, in the same way as in Experiment 3. In this case, a type-2 FLC was used and the results obtained with a type-1 FLC (Experiment 3) were improved. We can appreciate from Figure 5.20, that the lower overshoot and the best settling times are reached using a type-2 FLC. In

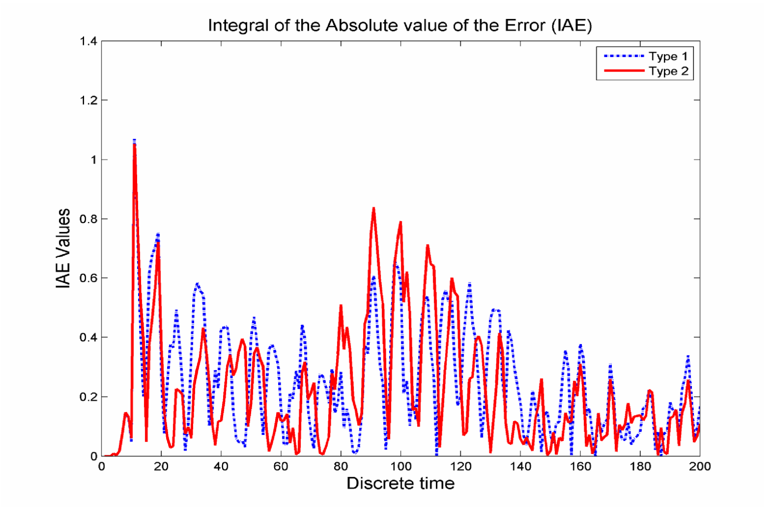


Fig. 5.22. In accordance with Fig. 5.20, IAE confirms that we obtained the best system response using a type-2 FLC with uncertainty presence. Moreover, the error of the settling time and steady state is lower using a type-2 FLC.

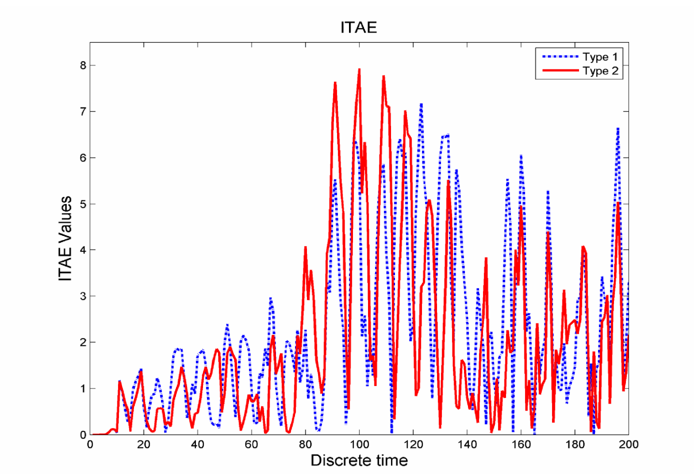


Fig. 5.23. Here we can see that the steady state error of the system produced by a type-2 FLC is lower than the error produced by a type-1 FLC with uncertainty present. ITAE will punish heavily all those errors produced with time.

Figures 5.21 and 5.22, we can see that with a type-2 FLC the overshoot error decreases very quickly and it remains lower than using a type-1 FLC. In Fig. 5.23, we can observe that through time the lower errors are obtained using a type-2 FLC.

Experiment 5: Varying the Signal to Noise Ratio (SNR) in type-1 and type-2 FLCs. To test the robustness of the type-1 and type-2 FLCs, we repeated experiments 3 and 4 giving different noise levels, going from 30 db to 8 db of SNR ratio in each experiment. In Table 5.4, we summarized the values for ISE, IAE, and ITAE considering 200 units of time with a P_{signal} of 22.98 db in all cases. As it can be seen in Table 5.4, in presence of different noise levels, the behavior of type-2 FLC is in general better than type-1 FLC.

Table 5.4. Behavior of type-1 and type-2 fuzzy logic controllers after variation of signal noise ratio. Values obtained for 200 samples.

Noise variation		**Type-1 FLC**			**Type-2 FLC**		
SNR (db)	Sum Noise (db)	ISE	IAE	ITAE	ISE	IAE	ITAE
8	22.72	321.1	198.1	2234.1	299.4	194.1	2023.1
10	20.762	178.1	148.4	1599.4	168.7	142.2	1413.5
12	18.783	104.7	114.5	1193.8	102.1	108.8	1057.7
14	16.785	64.1	90.5	915.5	63.7	84.8	814.6
16	14.78	40.9	72.8	710.9	40.6	67.3	637.8
18	12.78	27.4	59.6	559.1	26.6	54.2	504.4
20	10.78	19.4	49.5	444.2	18.3	44.8	402.9
22	8.78	14.7	42	356.9	13.2	37.8	324.6
24	6.78	11.9	36.2	289	10.3	32.5	264.2
26	4.78	10.1	31.9	236.7	8.5	28.6	217.3
28	2.78	9.1	28.5	196.3	7.5	25.5	180.7
30	0.78	8.5	25.9	164.9	7	23.3	152.6

From Table 5.4, considering two examples, the extreme cases; we have for an SNR ratio of 8 db, in type-1 FLC the following performance values ISE=321.1, IAE=198.1, ITAE=2234.1; and for the same case, in type-2 FLC, we have ISE=299.4, IAE=194.1, ITAE=2023.1.

For 30 db of SNR ratio, we have for the type-1 FLC, ISE=8.5, IAE=25.9, ITAE=164.9, and for the type-2 FLC, ISE=7, IAE=23.3, ITAE=152.6.

These values indicate a better performance of the type-2 FLC than type-1 FLC, because they are a representation of the errors, and as the error increases the performance of the system goes down.

5.4 Summary

We have presented the study of the controllers' design for nonlinear control systems using type-1 and type-2 fuzzy logic. We presented five experiments where we simulated the systems' responses with and without uncertainty presence. In the experiments, a quantification of errors was achieved and documented in tables for different criteria such as ISE, IAE, and ITAE, it was shown that the lower overshoot errors and the best settling times were obtained using a type-2 FLC. Based on the experimental results, we can say that the best results are obtained using type-2 fuzzy systems. In our opinion, this is because type-2 fuzzy sets that are used in type-2 fuzzy systems can handle uncertainties in a better way because they provide us with more parameters and more design degrees of freedom.

6 Method for Response Integration in Modular Neural Networks with Type-2 Fuzzy Logic

We describe in this chapter a new method for response integration in modular neural networks using type-2 fuzzy logic. The modular neural networks were used in human person recognition. Biometric authentication is used to achieve person recognition. Three biometric characteristics of the person are used: face, fingerprint, and voice. A modular neural network of three modules is used. Each module is a local expert on person recognition based on each of the biometric measures. The response integration method of the modular neural network has the goal of combining the responses of the modules to improve the recognition rate of the individual modules. We show in this chapter the results of a type-2 fuzzy approach for response integration that improves performance over type-1 fuzzy logic approaches.

6.1 Introduction

Today, a variety of methods and techniques are available to determine unique identity, the most common being fingerprint, voice, face, and iris recognition (Melin and Castillo, 2005). Of these, fingerprint and iris offer a very high level of certainty as to a person's identity, while the others are less exact. A large number of other techniques are currently being examined for suitability as identity determinants. These include (but are not limited to) retina, gait (walking style), typing style, body odour, signature, hand geometry, and DNA. Some wildly esoteric methods are also under development, such as ear structure, thermal imaging of the face and other parts of the body, subcutaneous vein patterns, blood chemistry, anti-body signatures, and heart rhythm, to name a few (Urias et al., 2006).

The four primary methods of biometric authentication in widespread use today are face, voice, fingerprint, and iris recognition. All of these are supported in our approach, some more abundantly than others. Generally, face and voice are considered to be a lower level of security than fingerprint and iris, but on the other hand, they have a lower cost of entry. We describe briefly in this section some of these biometric methods.

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 77–86, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

Face Recognition. Facial recognition has advanced considerably in the last 10 to 15 years. Early systems, based entirely on simple geometry of key facial reference points, have given way to more advanced mathematically-based analyses such as Local Feature Analysis and Eigenface evaluation. These have been extended though the addition of "learning" systems, particularly neural networks.

Face recognition systems are particularly susceptible to changes in lighting systems. For example, strong illumination from the side will present a vastly different image to a camera than neutral, evenly-positioned fluorescent lighting. Beyond this, however, these systems are relatively immune to changes such as weight gain, spectacles, beards and moustaches, and so on. Most manufacturers of face recognition systems claim false accept and false reject rates of 1% or better.

Voice Recognition. Software systems are rapidly becoming adept at recognising and converting free-flowing speech to its written form. The underlying difficulty in doing this is to flatten out any differences between speakers and understand everyone universally. Alternatively, when the goal is to specifically identify one person in a large group by their voice alone, these very same differences need to be identified and enhanced.

As a means of authentication, voice recognition usually takes the form of speaking a previously-enrolled phrase into a computer microphone and allowing the computer to analyse and compare the two sound samples. Methods of performing this analysis vary widely between vendors. None is willing to offer more than cursory descriptions of their algorithms--principally because, apart from LAN authentication, the largest market for speaker authentication is in verification of persons over the telephone.

Fingerprint Recognition. The process of authenticating people based on their fingerprints can be divided into three distinct tasks. First, you must collect an image of a fingerprint; second, you must determine the key elements of the fingerprint for confirmation of identity; and third, the set of identified features must be compared with a previously-enrolled set for authentication. The system should never expect to see a complete 1:1 match between these two sets of data. In general, you could expect to couple any collection device with any algorithm, although in practice most vendors offer proprietary, linked solutions.

A number of fingerprint image collection techniques have been developed. The earliest method developed was optical: using a camera-like device to collect a high-resolution image of a fingerprint. Later developments turned to silicon-based sensors to collect an impression by a number of methods, including surface capacitance, thermal imaging, pseudo-optical on silicon, and electronic field imaging.

As discussed, a variety of fingerprint detection and analysis methods exist, each with their own strengths and weaknesses. Consequently, researchers vary widely on their claimed (and achieved) false accept and false reject rates. The poorest systems offer a false accept rate of around 1:1,000, while the best are approaching 1:1,000,000. False reject rates for the same vendors are around 1:100 to 1:1000.

6.2 Proposed Approach for Recognition

Our proposed approach for human recognition consists in integrating the information of the three main biometric parts of the person: the voice, the face, and the fingerprint (Urias et al., 2006). Basically, we have an independent system for recognizing a person from each of its biometric information (voice, face, and fingerprint), and at the end we have an integration unit to make a final decision based on the results from each of the modules. In Figure 6.1 we show the general architecture of our approach in which it is clearly seen that we have one module for voice, one module for face recognition, and one module for fingerprint recognition. At the top, we have the decision unit integrating the results from the three modules. In this paper the decision unit is implemented with a type-2 fuzzy system.

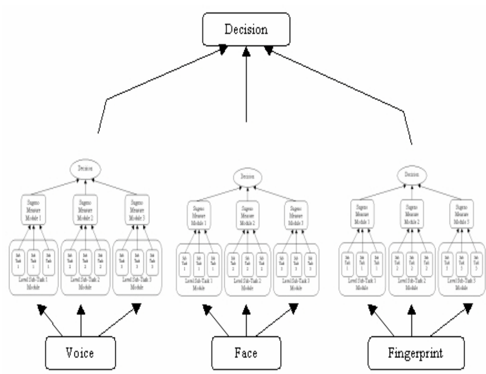


Fig. 6.1. Architecture of the proposed modular approach

6.3 Modular Neural Networks

This section describes a particular class of "modular neural networks", which have a hierarchical organization comprising multiple neural networks; the architecture basically consists of two principal components: local experts and an integration unit, as illustrated in Figure 6.2. In general, the basic concept resides in the idea that combined (or averaged) estimators may be able to exceed the limitation of a single estimator (Fogelman-Soulie, 1993). The idea also shares conceptual links with the "divide and conquer" methodology. Divide and conquer algorithms attack a complex problem by dividing it into simpler problems whose solutions can be combined to yield a solution to the complex problem (Monrocq, 1993). When using a modular network, a given task is split up among several local experts NNs (Happel and Murre, 1994). The average load on each NN is reduced in comparison with a single NN that must learn the entire original task, and thus the combined model may be able to surpass the limitation of a single NN. The outputs of a certain number of local experts (O_i) are mediated

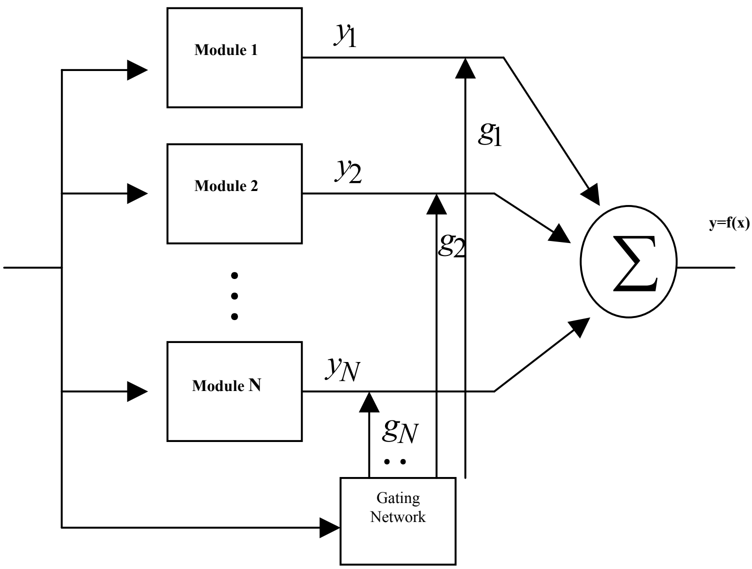


Fig. 6.2. Architecture of a modular neural network

by an integration unit. The integrating unit puts the outputs together using estimated combination weights (g_i). The overall output Y is given by equation (6.1).

$$Y_i = \Sigma g_i \, O_I \tag{6.1}$$

Nowlan, Jacobs, Hinton, and Jordan (Nowlan et al., 1991) described modular networks from a competitive mixture perspective. That is, in the gating network, they used the "softmax" function, which was introduced by (McCullagh and Nelder, 1994). More precisely, the gating network uses a softmax activation g_i of ith output unit given by

$$G_i = \exp\,(ku_i)/\ \Sigma_j \exp\,(ku_j) \tag{6.2}$$

Where u_i is the weighted sum of the inputs flowing to the ith output neuron of the gating network. Use of the softmax activation function in modular networks provides a sort of "competitive" mixing perspective because the ith local expert's output O_i with a minor activation u_i does not have a great impact on the overall output Y_i.

6.4 Integration of Results for Person Recognition Using Fuzzy Logic

On the past decade, fuzzy systems have displaced conventional technology in different scientific and system engineering applications, especially in pattern recognition and control systems. The same fuzzy technology, in approximation reasoning form, is resurging also in the information technology, where it is now giving support to decision making and expert systems with powerful reasoning capacity and a limited quantity of rules (Zadeh, 1998). For the case of modular neural networks, a fuzzy system can be used as an integrator or results (Melin and Castillo, 2005).

The fuzzy sets were presented by L. A. Zadeh in 1965 to process / manipulate data and information affected by unprobabilistic uncertainty / imprecision (Zadeh, 1975). These were designed to mathematically represent the vagueness and uncertainty of linguistic problems; thereby obtaining formal tools to work with intrinsic imprecision in different type of problems; it is considered a generalization of the classic set theory.

Type-2 fuzzy sets are used for modeling uncertainty and imprecision in a better way. These type-2 fuzzy sets were originally presented by Zadeh in 1975 and are essentially "fuzzy fuzzy" sets where the fuzzy degree of membership is a type-1 fuzzy set (Zadeh, 1996). The new concepts were introduced by (Mendel, 2001) allowing the characterization of a type-2 fuzzy set with a superior membership function and an inferior membership function; these two functions can be represented each one by a type-1 fuzzy set membership function. The interval between these two functions represent the footprint of uncertainty (FOU), which is used to characterize a type-2 fuzzy set. The uncertainty is the imperfection of knowledge about the natural process or natural state. The statistical uncertainty is the randomness or error that comes from different sources as we use it in a statistical methodology (Castillo et al., 2005).

6.5 Modular Neural Networks with Type-2 Fuzzy Logic as a Method for Response Integration

As was mentioned previously, type-2 fuzzy logic was used to integrate the responses of the three modules of the modular network. Each module was trained with the corresponding data, i.e. face, fingerprint and voice. Also, a set of modular neural networks was built to test the type-2 fuzzy logic approach of response integration. The architecture of the modular neural network is shown in Figure 6.3. From this figure we can appreciate that each module is also divided in three parts with the idea of also dividing each of the recognition problems in three parts.

Experiments were performed with sets of 20 and 30 persons. The trainings were done with different architectures, i.e. different number of modules, layers and nodes.

As can be appreciated from Figure 6.3, the first module was used for training with voice data. In this case, three different words were used for each person. The words used were: access, presentation, and hello.

The second module was used for training with person face data. In this case, two different photos were taken from each person, one in a normal position and the other with noise. The idea is that training with noise will make the recognition more robust to changes in the real world. We show in Figure 6.4 the photos of two persons in a normal situation and in a noisy situation.

The third module was used with fingerprint data of the group of persons. The fingerprint information was taken with a scanner. Noise was added for training the neural networks.

In all cases, each module is subdivided in three submodules, in this way making easier the respective recognition problem.

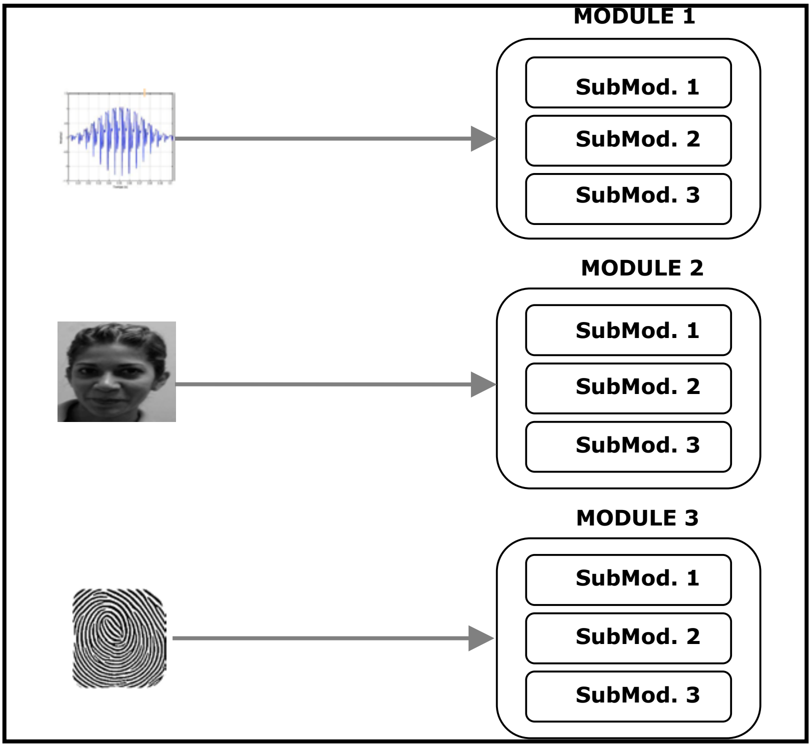


Fig. 6.3. Architecture of the Modular Network used for the recognition problem

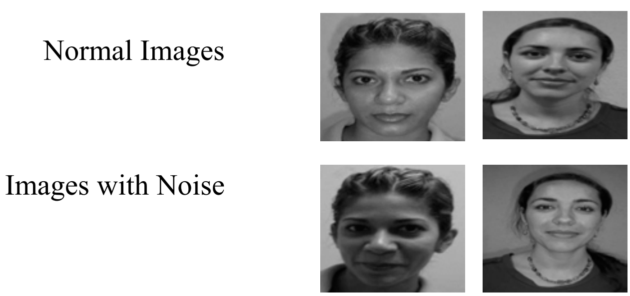


Fig. 6.4. Sample Photos of Faces in a Normal and Noisy Situation

6.6 Simulation Results

A set of different trainings for the modular neural networks was performed to test the proposed type-2 fuzzy logic approach for response integration in modular neural networks. We show in Table 1 some of these trainings with different numbers of modules, layers and nodes. The training times are also shown in this table to illustrate the performance with different training algorithms and conditions.

Table 6.1. Sample Trainings of the Modular Neural Network

Red	Funcion de Entrenamiento	Num. de Capas	Neuronas	F. R.	% de Reconocimiento	Error Meta	Error Alcanzado	Epocas	Time
1	Trainscg	V: Mod1: 2	48,49	sse	100% (20/20)	0.001	0.000998658	3000	35 Min.
		Mod2: 2	50,60	sse	100% (20/20)	0.001	0.000998254	3000	
		Mod3: 2	60,70	sse	100% (20/20)	0.001	0.00099876	3000	
		R: Mod4: 1	350	mse	100% (20/20)	0.01	0.0099726	4000	
		Mod5: 1	400	mse	100%(20/20)	0.01	0.0099546	4000	
		Mod6: 1	420	mse	100% (20/20)	0.01	0.0098316	4000	
		H: Mod7: 1	350	mse	100% (20/20)	0.01	0.0080223	4000	
		Mod8: 1	250	mse	100% (20/20)	0.01	0.0086453	4000	
		Mod9: 1	300	mse	100% (20/20)	0.01	0.0067892	4000	

Red	Funcion de Entrenamiento	Num. de Capas	# de Neuronas	F.R.	% de Reconocimiento	Error Meta	Error Alcanzado	Epocas	Time
10	Trainscg	V: Mod1: 2	80,90	sse	100% (20/20)	0.001	0.000999948	3000	1 Hra. 34 Min.
		Mod2: 2	90,90	sse	100% (20/20)	0.001	0.000992595	3000	
		Mod3: 2	80,90	sse	100% (20/20)	0.001	0.000997131	3000	
	Trainscg	R: Mod4: 1	20	mse	100% (20/20)	0.01	0.124015	4000	
		Mod5: 1	15	mse	100% (20/20)	0.01	0.0269689	4000	
		Mod6: 1	25	mse	100% (20/20)	0.01	0.01698	4000	
	Traingdx	H: Mod7: 1	350	mse	25% (5/20)	0.01	0.037501	4000	
		Mod8: 1	230	mse	5% (1/20)	0.01	0.0450007	4000	
		Mod9: 1	290	mse	5% (1/20)	0.01	0.0425011	4000	

Red	Funcion de Entrenamiento	Num. de Capas	# de Neuronas	F.R.	% de Reconocimiento	Error Meta	Error Alcanzado	Epocas	Time
14	Trainscg	V: Mod1: 2	85,95	mse	100% (30/30)	0.001	0.000990206	3000	1 Hra. 23 Min.
		Mod2: 2	95,90	mse	100% (30/30)	0.001	0.000970259	3000	
		Mod3: 2	99,96	mse	93% (28/30)	0.001	0.000995249	3000	
	Trainscg	R: Mod4: 1	25	mse	0.3% (1/30)	0.01	0.0880936	4000	
		Mod5: 1	20	mse	0.3% (1/30)	0.01	0.0172602	4000	
		Mod6: 1	30	mse	0.3% (1/30)	0.01	0.0127059	4000	
	Traincsg	H: Mod7: 1	15	mse	96% (29/30)	0.01	0.148242	4000	
		Mod8: 1	10	mse	90% (27/30)	0.01	0.142394	4000	
		Mod9: 1	23	mse	96% (29/30)	0.01	0.127423	4000	

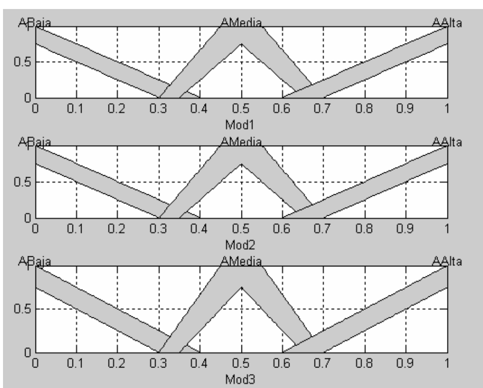


Fig. 6.5. Input variables of the type-2 fuzzy system

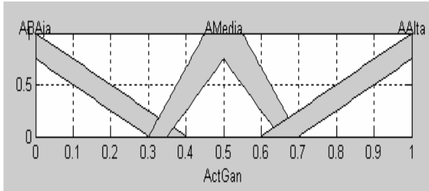


Fig. 6.6. Output variables of the type-2 fuzzy system

Table 6.2. Results of the Type-2 Fuzzy System with Triangular Membership Functions

Funciones de Membresia Triangulares(7)	
Entrenamiento	**% de Reconocimiento**
1	100% (20/20)
2	100% (20/20)
3	100% (20/20)
4	100% (20/20)
5	100% (20/20)
6	5% (1/20)
7	100% (20/20)
8	65% (13/20)
9	100% (20/20)
10	100% (20/20)
11	93% (28/30)
12	96% (29/30)
13	93% (28/30)
14	93% (28/30)
15	83% (25/30)
Resultados para cada uno de los entrenamientos , utilizando un Sistema Difuso con Funciones de MembresiaTriangulares	

Once the necessary trainings were done, a set of tests were performed with different type-2 fuzzy systems. The fuzzy systems were used as response integrators for the three modules of the modular network. In the type-2 fuzzy systems, different types of membership functions were considered with goal of comparing the results and deice on the best choice for the recognition problem.

The best type-2 fuzzy system, in the sense that it produced the best recognition results, was the one with triangular membership functions. This fuzzy system has 3 input variables and one output variable, with three membership functions per variable. We show in Figures 6.5 and 6.6 the membership functions of the type-2 fuzzy system.

The recognition results of this type-2 fuzzy system for each training of the modular neural network are shown in Table 2.

In Table 6.2 we show the results for 15 trainings of the modular neural network. In each row of this table we can appreciate the recognition rate with the type-2 fuzzy system. We can appreciate that in 8 out of 15 cases, a 100% recognition rate was achieved.

The fuzzy systems with worst results for the modular neural network were the ones with Gaussian and Trapezoidal membership functions. We use 3 input variables and one output variable, as in the previous fuzzy system. We show in Figures 6.7 and 6.8 the Gaussian membership functions of this system.

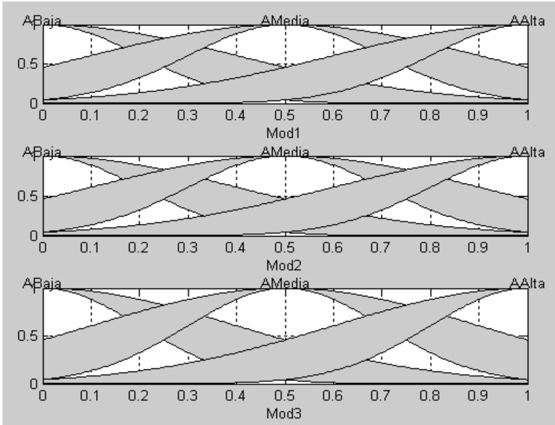


Fig. 6.7. Input variables for type-2 fuzzy system with Gaussian membership functions

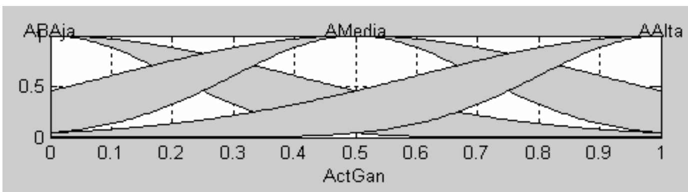


Fig. 6.8. Output variable for type-2 fuzzy system with Gaussian membership functions

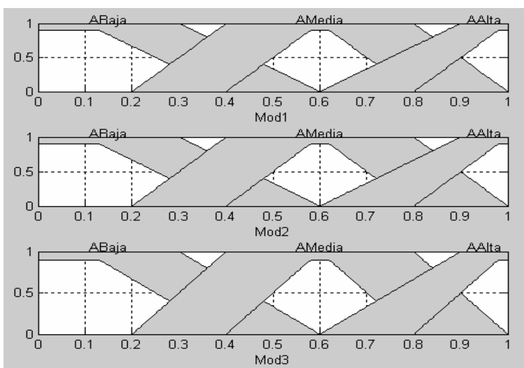


Fig. 6.9. Input variables for the Type-2 Fuzzy System with Trapezoidal Functions

We show in Figures 6.9 and 6.10 the Trapezoidal membership functions of another type-2 fuzzy system.

The results that were obtained with Gaussian and Trapezoidal membership functions are similar. We show in Table 3 the recognition results obtained with the type-2 fuzzy system with Trapezoidal membership functions. We can appreciate from Table 6.3

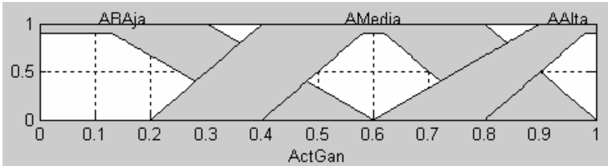


Fig. 6.10. Output variable for type-2 fuzzy system with Trapezoidal functions

Table 6.3. Recognition rates with the Type-2 System and Trapezoidal Functions

Funciones de Membresia Trapezoidales (11)	
Entrenamiento	**% de Reconocimiento**
1	100% (20/20)
2	100% (20/20)
3	55% (11/20)
4	100% (20/20)
5	100% (20/20)
6	95% (19/20)
7	100% (20/20)
8	60% (12/20)
9	55% (11/20)
10	45% (9/20)
11	96% (29/30)
12	96% (29/30)
13	100% (30/30)
14	6% (2/30)
15	3% (1/30)
Resultados para cada uno de los entrenamientos, utilizando un Sistema Difuso con Funciones de Membresia Trapezoidal	

that only in 6 out of the 15 cases a 100% recognition rate is obtained. Also, there are 4 cases with low recognition rates.

We have to mention that results with a type-1 fuzzy integration of responses were performed in previous paper, in which the recognition rates were consistently lower by an average of 5%. We can state in conclusion that the type-2 fuzzy system for response integration is improving the recognition rate in the case of persons based on face, fingerprint and voice.

6.7 Summary

We described in this chapter a new method for response integration in modular neural networks that uses type-2 fuzzy logic to model uncertainty in the decision process. We showed different trainings of the modular neural networks, and tested different type-2 fuzzy systems for response integration. Based on the obtained recognition rates, the best results were achieved with a type-2 fuzzy system with triangular membership functions. The results obtained with this type-2 fuzzy system are better than the previously obtained by a similar type-1 approach.

7 Type-2 Fuzzy Logic for Improving Training Data and Response Integration in Modular Neural Networks for Image Recognition

The combination of Soft Computing techniques allows the improvement of intelligent systems with different hybrid approaches. In this work we consider two parts of a Modular Neural Network for image recognition, where a Type-2 Fuzzy Inference System (FIS 2) makes a great difference. The first FIS 2 is used for feature extraction in training data, and the second one to find the ideal parameters for the integration method of the modular neural network. Once again Fuzzy Logic is shown to be a tool that can help improve the results of a neural system, when facilitating the representation of the human perception.

7.1 Method for Image Recognition

At the moment, many methods for image recognition are available. But most of them include a phase of feature extraction or another type of preprocessing closely related to the type of image to recognize (Melin and Castillo, 2005) (Starovoitov et al., 2002) (Chuang et al., 2000). The method proposed in this paper can be applied to any type of images, because the preprocessing phase does not need specific data about the type of image (Melin et al., 2007) (Mendoza and Melin, 2007).

Even if the method was not designed only for face recognition, we have made the tests with the ORL face database (AT&T Laboratories Cambridge) composed of 400 images of size 112x92. There are 40 persons, with 10 images of each person. The images are taken at different times, lighting and facial expressions. The faces are in upright position of frontal view, with slight left-right rotation. Figure 7.1 shows the 10 samples of one person in ORL database.

To explain the proposed steps of the method, we need to separate it them in two phases: the training phase in figure 7.3 and the recognition phase in figure 7.4.

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 87–94, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

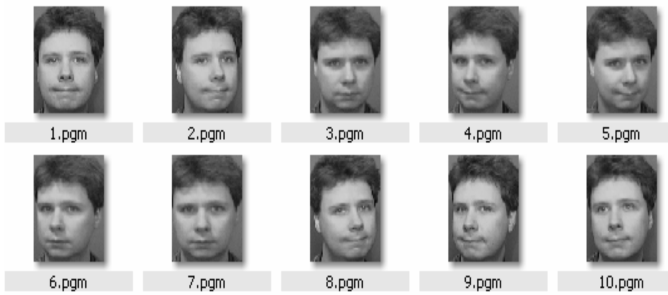


Fig. 7.1. Set of 10 samples of a person in ORL

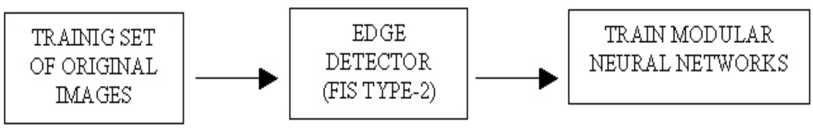


Fig. 7.2. Steps in Training Phase

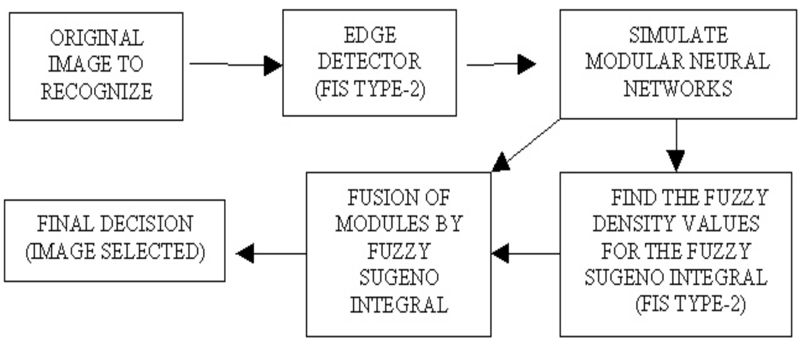


Fig. 7.3. Steps in Recognition Phase

7.2 Type-2 Fuzzy Inference System as Edge Detector

In previous work we presented an efficient Fuzzy Inference System for edges detection, in order to use the output image like input data for modular neural networks (Mendoza and Melin, 2006). In the proposed technique, it is necessary to apply Sobel operators to the original images, then use a Fuzzy Inference System Type-2 to generate the vector of edges that would serve like input data in a neural network. Type-2 Fuzzy Logic enables us to handle uncertainties in decision making and recognition in a more convenient way and for this reason was proposed (Castillo et al., 2007).

For the Type-2 Fuzzy Inference System, 3 inputs are required, 2 of them are the gradients with respect to x-axis and y-axis, calculated with (1), to which we will call DH and DV respectively.

The Sobel edges detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{7.1}$$

Where Sobely y Sobelx are the Sobel Operators throughout x-axis and y-axis.

If we define I as the source image, g_x and g_y are two images which at each point contain the horizontal and vertical derivative approximations, the latter are computed as (2) and (3).

$$g_x = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{x,i,j} * I_{r+i-2,c+j-2} \tag{7.2}$$

$$g_y = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{y,i,j} * I_{r+i-2,c+j-2} \tag{7.3}$$

Where gx and gy are the gradients along axis-x and axis-y, and * represents the convolution operator.

The other input is a filter that calculates when applying a mask by convolution to the original image. The low-pass filter hMF (4) allow us to detect image pixels belonging to regions of the input were the mean gray level is lower. These regions are proportionally more affected by noise, supposed it is uniformly distributed over the whole image.

The goal here is to design a system which makes it easier to include edges in low contrast regions, but which does not favor false edges by effect of noise (Miosso and Bauchspiess, 2001).

$$hMF = \frac{1}{25} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{7.4}$$

Then the inputs for FIS type 2 are: DH=g_x, DV=g_y, M= $hMF*I$, where * is the convolution operator, and de output is a column vector contains the values of the image edges, and we can represent that in graphics shown in figure 7.4.

The Edges Image is smaller than the original because the result of convolution operation is a central matrix where the convolution has a value. Then in our example, each image with dimension 112x92 is reduced to 108x88.

The inference rules and membership function parameters allow to calculate a gray value between -4.5 and 1.5 for each pixel, where the most negative values corresponds

corresponds to the dark tone in the edges of the image. Then if we see the rules, only when the increment value of the inputs DH and DV are low the output is HIGH or clear (the background), in the rest of rules the output is LOW or dark (the edges). The complete set of fuzzy rules is given as follows (Castro et al., 2006):

1. If (DH is LOW) and (DV is LOW) then (EDGES is HIGH) (1)
2. If (DH is MEDIUM) and (DV is MEDIUM) then (EDGES is LOW) (1)
3. If (DH is HIGH) and (DV is HIGH) then (EDGES is LOW) (1)
4. If (M is LOW) and (DV is MEDIUM) then (EDGES is LOW) (1)
5. If (M is LOW) and (DH is MEDIUM) then (EDGES is LOW) (1)

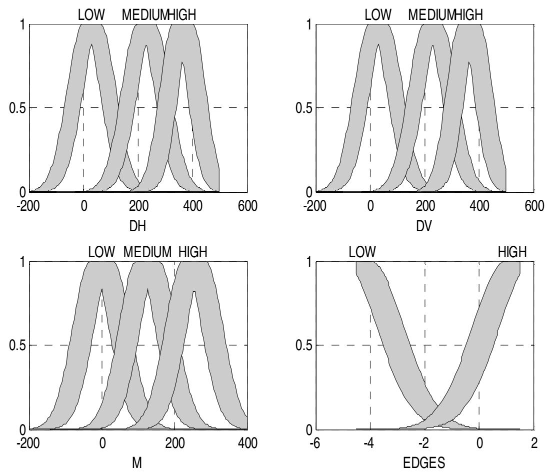


Fig. 7.4. Membership Function for the Type-2 FIS Edge Detector

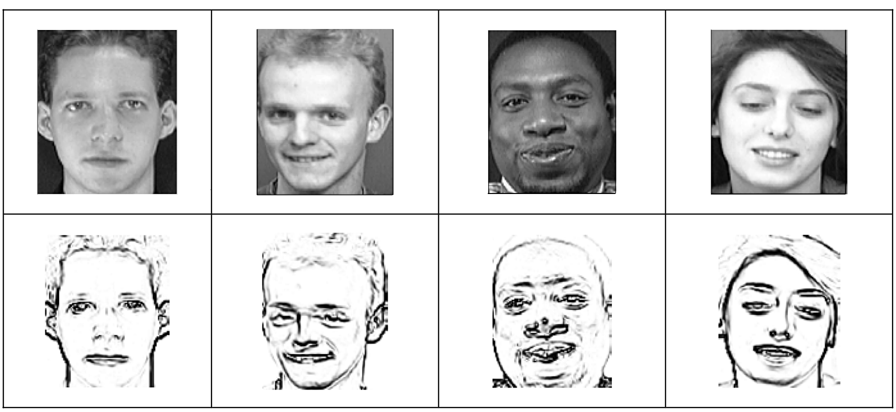


Fig. 7.5. Examples of edge detection with the Type-2 FIS method

The edge detector allows us to ignore the background color. We can see in this database of faces, different tones present for the same or another person. Then we eliminate a possible influence of a bad classification by the neural network, without losing detail in the image. Another advantage of edge detector is that the values can be normalized to a homogenous value range, independently the light, contrast or background tone in each image. At the examples in figure 7.5, all the edges in the images have a minimum value of -3.8 and a maximum value of 0.84. In particular for neural network training, we find these values to make the training faster: the mean of the values is near 0 and the standard deviation is near 1 for all the images.

7.3 The Modular Structure

The design of the Modular Neural Network consists of 3 monolithic feedforward neural networks (Sharkey, 1999), each one trained with a supervised method with the first 7 samples of the 40 images. Then the edges vector column is accumulated until the number of samples to form the input matrix for the neural networks as it is in the scheme of figure 7.7. Once the complete matrix of images is divided in 3 parts, each module is training with a correspondent part, with some rows of overlap.

The target to the supervised training method consist of one identity matrix for each sample, building one matrix with dimensions 40x(40*number_of_samples), as shown in figure 7.8.

Each Monolithic Neural Network has the same structure and is trained under the same conditions, like we can see in the next code segment:

```
layer1=200; layer2=200;
layer3=number_of_subjects;
net=newff(minmax(p),[layer1,layer2,layer3],{'tan
sig','tansig','logsig'},'traingdx');
net.trainParam.goal=1e-5;
net.trainParam.epochs=1000;
```

The average number of epochs to meet the goal in each module is of 240, and the required time of 160 seconds.

7.4 Simulation Results

A program was developed in Matlab that simulates each module with the 400 images of the ORL database, building a matrix with the results of the simulation of each module. These matrices are stored in the file "mod.mat" to be analyzed later for the combination of results. We can observe that in the columns corresponding to the training data, the position with a value near one is the image selected correctly. However in the columns that correspond to the test data this doesn't always happens, reason why it is very important to have a good combination method to recognize more images.

According to exhaustive tests made in the simulation matrices, we know that recognition of the images that were used for the training of the neural networks is of the

100%. Therefore the interest is focused on the recognition of the samples that do not belong to the training set, is to say samples 8,9 and 10. The parameters for the Sugeno Fuzzy Integral that will be inferred will be the Fuzzy Densities, a value between 0 and 1 for each module, which determines the rate for each module. The parameter lambda, according to the theory of fuzzy measures depends on the values of the fuzzy densities, and is calculated by searching for the roots of a polynomial. After the simulation of an image in the Neural Network, the simulation value is the only known parameter to make a decision, then to determine the fuzzy density for each module is the unique available information. For this reason we analyze the values in many simulations matrix and decide that each input to the FIS Type-2 corresponds to the maximum value of each column corresponding to the simulation of each module of each one of the 400 images. The process to recognize each one of the images is shown in figure 6.

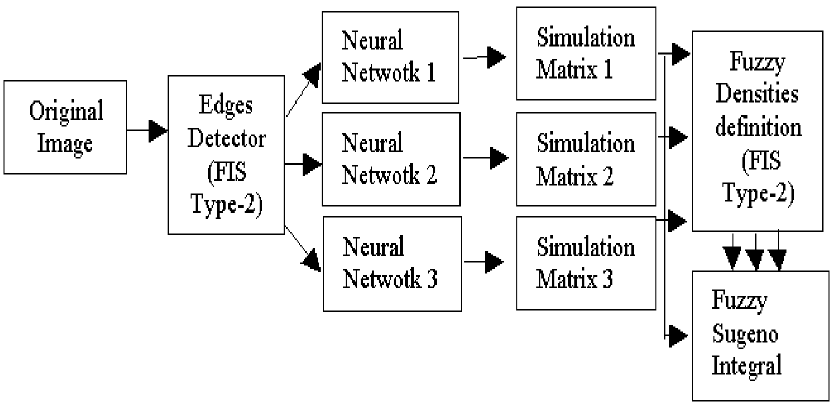


Fig. 7.6. Process of recognition using the type-2 fuzzy modular approach

Then each output corresponds to one fuzzy density, to be applied for each module to perform the fusion of results later with the Fuzzy Sugeno Integral. The inference rules found fuzzy densities near 1 when de maximum value in the simulation is between 0.5 and 1, and near 0 when the maximum value in the simulation is near 0. The fuzzy rules are shown below and membership functions in Figure 7.7.

1. If (max1 is LOW) then (d1 is LOW) (1)
2. If (max2 is LOW) then (d2 is LOW) (1)
3. If (max3 is LOW) then (d3 is LOW) (1)
4. If (max1 is MEDIUM) then (d1 is HIGH) (1)
5. If (max2 is MEDIUM) then (d2 is HIGH) (1)
6. If (max3 is MEDIUM) then (d3 is HIGH) (1)
7. If (max1 is HIGH) then (d1 is HIGH) (1)
8. If (max2 is HIGH) then (d2 is HIGH) (1)
9. If (max3 is HIGH) then (d3 is HIGH) (1)

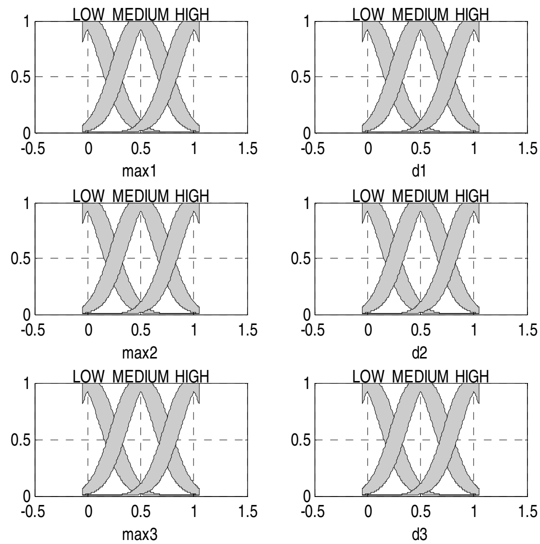


Fig. 7.7. Membership functions for the FIS to find fuzzy densities

Although the rules are very simple, allows to model the fuzziness to rate de modules when the simulation result don't reach the maximum value 1.

However some of the images don't reach the sufficient value in the simulation of the three modules, in these cases, do not exists enough information to select an image at the modules combination, and the image is wrongly selected.

In order to measure of objective form the final results, we developed a method of random permutation, which rearranges the samples of each person before the training. Once a permutation is made, the modular neural networks are trained and combined four times to obtain the sufficient information to validate the results. The average recognition rate is of 96.5%.

We show in Table 7.1 the summary of simulation results for each of the modules and the average and maximum results of the modular network (after fusion or combination of the results).

Table 7.1. Summary of the simulation results with the hybrid approach

Permu-tation	Image Recognition (%)					
	Train 1	Train 2	Train 3	Train 4	Average	Maximum
1	92.75	95	92.2	93.25	93.3	95
2	96.5	95.25	94.25	95.5	95.375	96.5
3	91.5	92	93.75	95.25	93.125	95.25
4	94.5	94.5	93.25	94	94.0625	94.5
5	93.75	93.5	94	96	94.3125	96
					94.035	**96.5**

7.5 Summary

We have shown in this chapter that the combination of Soft Computing techniques allows the improvement of intelligent systems with different hybrid approaches. In this chapter we considered two parts of a Modular Neural Network for image recognition, where a Type-2 Fuzzy Inference System (FIS 2) help us improves the performance results in image recognition. The first FIS 2 was used for feature extraction in training data, and the second one to find the ideal parameters for the integration method of the modular neural network.

8 Fuzzy Inference Systems Type-1 and Type-2 for Digital Images Edge Detection

Edges detection in digital images is a problem that has been solved by means of the application of different techniques from digital signal processing, also the combination of some of these techniques with Fuzzy Inference System (FIS) has been experienced. In this chapter a new FIS Type-2 method is implemented for the detection of edges and the results of three different techniques for the same intention are compared.

8.1 Introduction

In the area of digital signal processing, methods have been proven that solve the problem of image recognition. Some of them include techniques like binarization, bidimensional filtrate, detection of edges and compression using banks of filters and trees, among others.

Specifically in methods for the detection of edges we can find comparative studies of methods like: Canny, Narwa, Iverson, Bergholm y Rothwell. Others methods can group in two categories: Gradient and Laplacian (Heath, 1996).

The gradient methods like Roberts, Prewitt and Sobel detect edges, looking for maximum and minimum in first derived from the image. The Laplacian methods like Marrs-Hildreth do it finding the zeros of second derived from the image (Mendoza and Melin, 2005).

This work is the beginning of an effort for the design of new pre-processing images techniques, using Fuzzy Inference Systems (FIS), that allows feature extraction and construction of input vectors for neural networks with aims of image recognition.

Artificial neural networks are one of the most used objective techniques in the automatic recognition of patterns, here some reasons:

- Theoretically any function can be determined.
- Except the input patterns, it is not necessary to provide additional information.
- They are possible to be applied to any type of patterns and to any data type.

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 95–107, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

The idea to apply artificial neuronal networks for images recognition, tries to obtain results without providing another data that the original images, of this form the process is more similar to the form in which the biological brain learns to recognize patterns, only knowing experiences of past.

Models with modular neural networks have been designed, that allow recognizing images divided in four or six parts. This is necessary due to the great amount of input data, since an image without processing is of 100x100 pixels, needs a vector 10000 elements, where each one corresponds to pixel with variations of gray tones between 0 and 255 (Mendoza and Melin, 2005).

This chapter shows an efficient Fuzzy Inference System for edges detection, in order to use the output image like input data for modular neural networks. In the proposed technique, it is necessary to apply Sobel operators to the original images, and then use a Fuzzy System to generate the vector of edges that would serve as input data to a neural network.

8.2 Sobel Operators

The Sobel operator applied on a digital image, in gray scale, calculates the gradient of the intensity of brightness of each pixel, giving the direction of the greater possible increase of black to white, in addition calculates the amount of change of that direction.

The Sobel operator performs a 2-D spatial gradient measurement on an image. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image.

The Sobel edges detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).

A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The Sobel masks are shown in equation (8.1) (Green, 2002):

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{8.1}$$

Where Sobely y Sobelx are the Sobel Operators throughout x-axis and y-axis.

If we define I as the source image, g_x and g_y are two images which at each point contain the horizontal and vertical derivative approximations, the latter are computed as in equations (8.2) and (8.3).

$$g_x = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{x,i,j} * I_{r+i-2,c+j-2} \tag{8.2}$$

$$g_y = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{y,i,j} * I_{r+i-2,c+j-2} \tag{8.3}$$

Where *gx* and *gy* are the gradients along axis-x and axis-y, and * represents the convolution operator.

The gradient magnitude *g* is calculated with equation (8.4) (Fan et al., 2004).

$$g = \sqrt{g_x^2 + g_y^2} \tag{8.4}$$

8.3 Edge Detection by Gradient Magnitude

Although the idea presented in this chapter, is to verify the efficiency of a FIS for edges detection in digital images, from the approaches given by Sobel operator, is necessary to display first the obtained results using only the gradient magnitude.

It will be used as an example the first image of the subject number one of the ORL database (figure 8.1). The gray tone of each pixel of this image is a value of between 0 and 255.

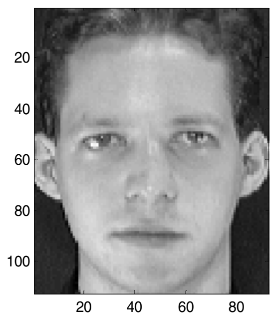


Fig. 8.1. Original Image 1.pgm

In figure 8.2 appears the image generated by *gx*, and figure 8.3 presents the image generated by *gy*.

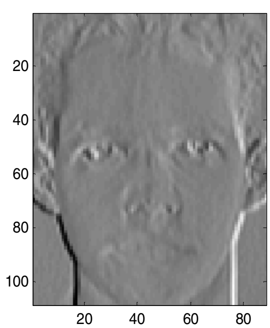


Fig. 8.2. Image given by *gx*

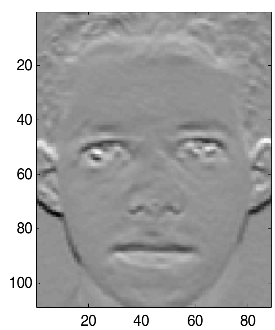


Fig. 8.3. Image given by *gy*

An example of maximum and minimum values of the matrix given by *gx*, *gy* and *g* from the image 1.pgm is shown in table 8.1.

Table 8.1. Maximum and minimum values from 1.pgm, gx, gy y g

Tone	1.pgm	gx	gy	g
Minimum	11	-725	-778	0
Maximum	234	738	494	792

After applying equation (8.4), g is obtained as it is in figure 8.4.

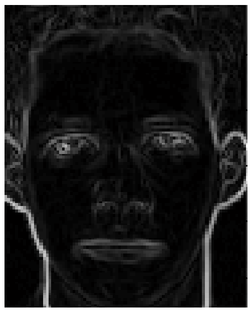


Fig. 8.4. Edges image given by g

8.4 Edge Detection Using Type-1 Fuzzy Logic

A Mamdani FIS was implemented using Type-1 Fuzzy Logic, with four inputs, one output and 7 rules, using the Matlab Fuzzy Logic Toolbox, which is shown in figure 8.5.

For the Type-1Fuzzy Inference System, 4 inputs are required, 2 of them are the gradients with respect to x-axis and y-axis, calculated with equation (2) and equation (3), to which we will call DH and DV respectively.

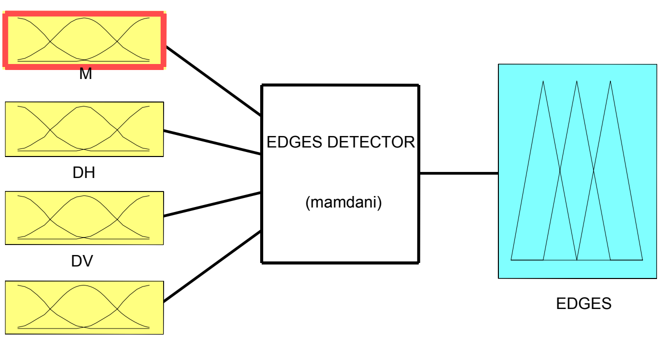


Fig. 8.5. FIS in Matlab Fuzzy Logic Tool Box

The other two inputs are filters: A high-pass filter, given by the mask of the equation (8.5), and a low-pass filter given by the mask of equation (8.6). The high-pass filter hHP detects the contrast of the image to guarantee the border detection in relative low contrast regions. The low-pass filter hMF allow to detects image pixels belonging to regions of the input were the mean gray level is lower. These regions are proportionally more affected by noise, supposed it is uniformly distributed over the whole image.

The goal here is to design a system which makes it easier to include edges in low contrast regions, but which does not favor false edges by effect of noise (Miosso and Bauchspiess, 2001).

$$hHP = \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix} \tag{8.5}$$

$$hMF = \frac{1}{25} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{8.6}$$

Then the inputs for FIS type 1 are:

DH=g_x
DV=g_y
HP= $hHP*I$
M= $hMF*I$

where * is the convolution operator.

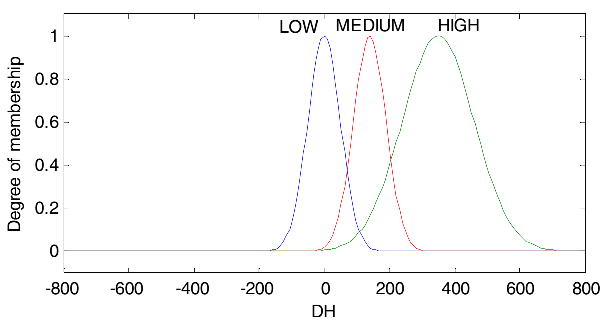


Fig. 8.6. Input variable DH

For all the fuzzy variables, the membership functions are of Gaussian type. According to the executed tests, the values in DH and DV, go from -800 to 800, then the ranks in x-axis adjusted as it is in figures 8.6, 8.7 and 8.8, in where the membership functions are:

LOW: gaussmf(43,0),
MEDIUM: gaussmf(43,127),
HIGH: gaussmf(43,255).

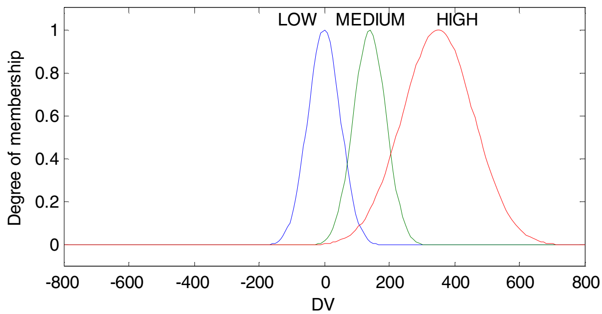


Fig. 8.7. Input variable DV

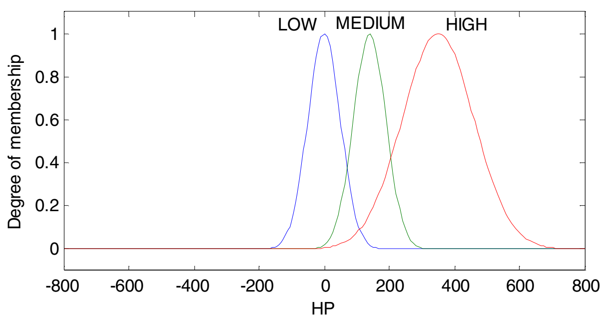


Fig. 8.8. Input variable HP

In the case of variable M, the tests threw values in the rank from 0 to 255, and thus the rank in x-axis adjusted, as it is appraised in figure 8.9.

In figure 8.10 is the output variable EDGES that also adjusted the ranks between 0 and 255, since it is the range of values required to display the edges of an image.
The seven fuzzy rules that allow to evaluate the input variables, so that the exit image displays the edges of the image in color near white (HIGH tone), whereas the background was in tones near black (tone LOW).

1. If (DH is LOW) and (DV is LOW) then (EDGES is LOW)
2. If (DH is MEDIUM) and (DV is MEDIUM) then (EDGES is HIGH)
3. If (DH is HIGH) and (DV is HIGH) then (EDGES is HIGH)

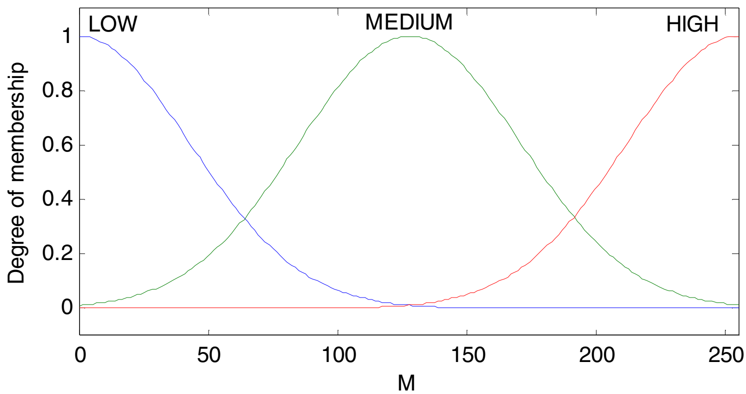


Fig. 8.9. Input variable M

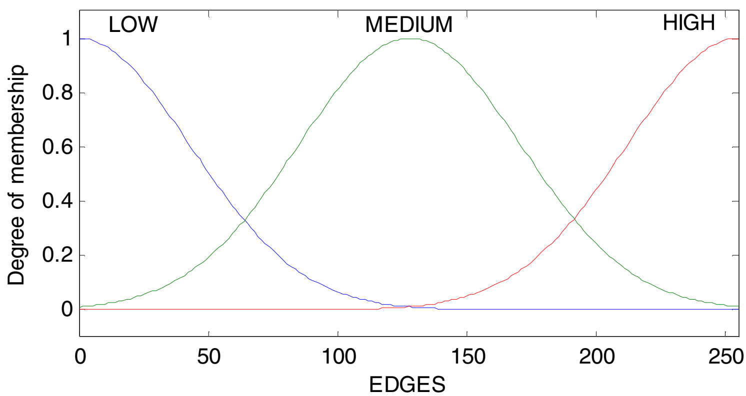


Fig. 8.10. Output variable EDGES

Fig. 8.11. EDGES Image by FIS Type 1

4. If (DH is MEDIUM) and (HP is LOW) then (EDGES is HIGH)
5. If (DV is MEDIUM) and (HP is LOW) then (EDGES is HIGH)
6. If (M is LOW) and (DV is MEDIUM) then (EDGES is LOW)
7. If (M is LOW) and (DH is MEDIUM) then (EDGES is LOW)

The result obtained for image of figure 1 is remarkably better than the one than it was obtained with the method of gradient magnitude, as it is in figure 8.11.

Reviewing the values of each pixel, we see that all fall in the rank from 0 to 255, which is not obtained with the method of gradient magnitude.

8.5 Edge Detection Using Type-2 Fuzzy Logic

For the Type-2 FIS, the same method was followed as in Type-1 FIS, indeed to be able to make a comparison of both results. The tests with the type-2 FIS, were executed using the computer program imagen_bordes_fis2.m, which creates a Type-2 Inference System (Mamdani) by intervals (Mendel, 2001).

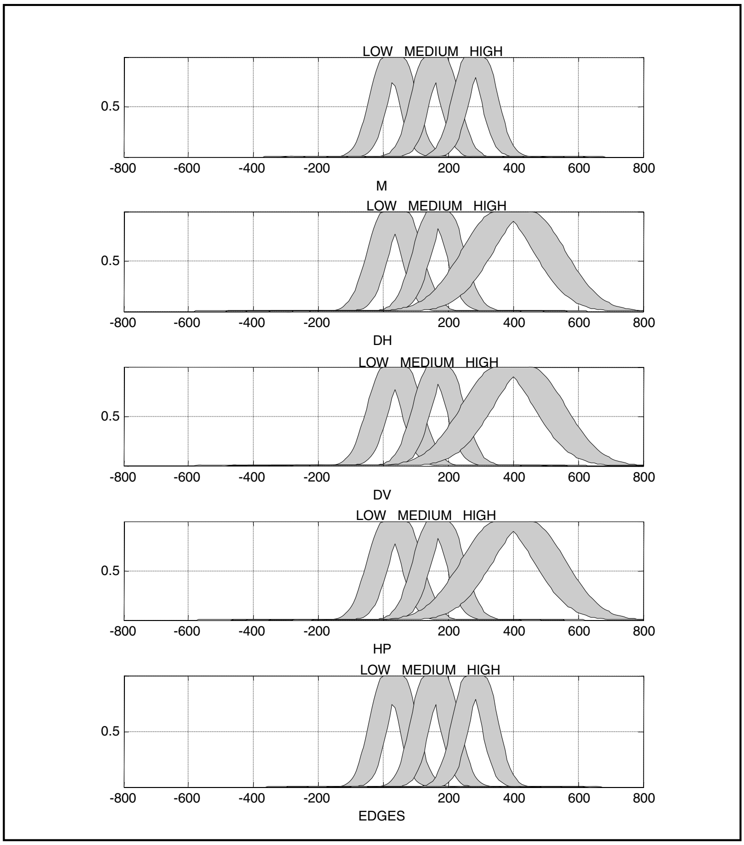


Fig. 8.12. Type-2 fuzzy variables

Fig. 8.13. EDGES Image by FIS Type 2

The mentioned program creates the fuzzy variables type 2 as it is seen in figure 8.12. The wide of the FOU chosen for each membership function was the one that had better results after several experiments.

The program imagen_bordes_fuzzy2.m was implemented to load the original image, and to apply the filters before mentioned. Because the great amount of data that the fuzzy rules must evaluate, the image was divided in four parts, and the FIS was applied to each one separately. The result of each evaluation gives a vector with tones of gray by each part of the image, in the end is the complete image with the edges (figure 8.13).

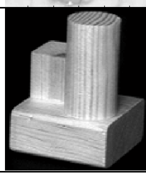
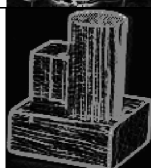
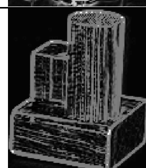
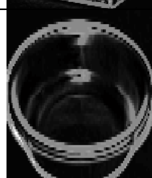
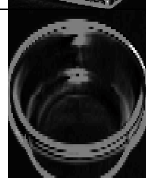
8.6 Comparison of Results

The first results of several tests conducted in different images can be appreciated in table 8.2.

At first, the results with FIS Type-1 and FIS Type2 are seen very similar. However thinking about that to show the images with a dark background it could confuse the contrast of tones, tests were done inverting the consequent of the rules, so that the edges take the dark tone and the bottom the clear tone, the rules changed to the following form:

1. If (DH is LOW) and (DV is LOW) then (EDGES is HIGH)
2. If (DH is MEDIUM) and (DV is MEDIUM) then (EDGES is LOW)
3. If (DH is HIGH) and (DV is HIGH) then (EDGES is LOW)
4. If (DH is MEDIUM) and (HP is LOW) then (EDGES is LOW)
5. If (DV is MEDIUM) and (HP is LOW) then (EDGES is LOW)
6. If (M is LOW) and (DV is MEDIUM) then (EDGES is HIGH)
7. If (M is LOW) and (DH is MEDIUM) then (EDGES is HIGH)

Table 8.2. Results of Edge Detection by FIS1 y FIS2
(dark background)

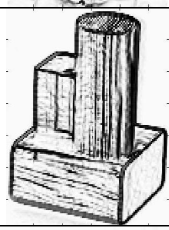
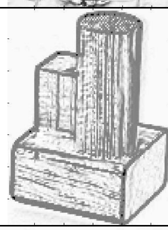
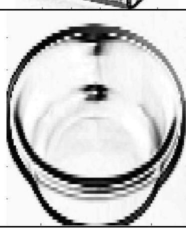
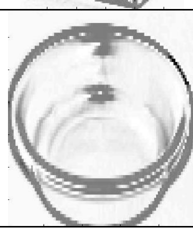
Original Image	EDGES (FIS 1)	EDGES (FIS 2)
		
		
		
		
		

Fuzzy Systems were tested both (Type-1 and Type-2), with the new fuzzy rules and same images, obtaining the results that are in table 8.3.

In this second test can be appreciated a great difference between the results obtained with the FIS 1 and FIS 2, noticing at first a greater contrast in the images obtained with the FIS 1 and giving to the impression of a smaller range of tones of gray in the type-2 FIS.

In order to obtain an objective comparison of the images, histograms were elaborated respectively [14] corresponding to the resulting matrices of edges of the FIS 1 and FIS 2, which are in table 8.4.

Table 8.3. Results of Edge Detection by FIS1 y FIS2
(clear background)

EDGES (FIS 1)	EDGES (FIS 2)
	
	
	
	

The histograms show in the y-axis the range of tones of gray corresponding to each image and in x-axis the frequency in which he appears pixel with each tone.

As we can observe, unlike detector FIS1, with FIS2 the edges of an image could be obtained from very complete form, only taking the tones around 150 and 255.

Like a last experiment, in this occasion to the resulting images of the FIS Type-2 the every pixel out of the range between 50 and 255 was eliminated.

Table 8.5 shows the amount of elements that was possible to eliminate in some of the images, we see that the Type-2 Edges Detector FIS allows to using less than half of the original pixels without losing the detail of the images. This feature could be a great advantage if these images are used like input data in neural networks for detection of images instead the original images.

Table 8.4. Histograms Of The Resulting Images Of The Edges by Gradient Magnitud, Fis 1 And Fis 2 Methods

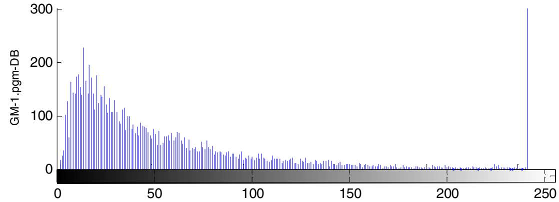
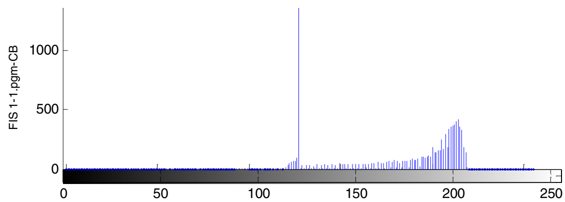
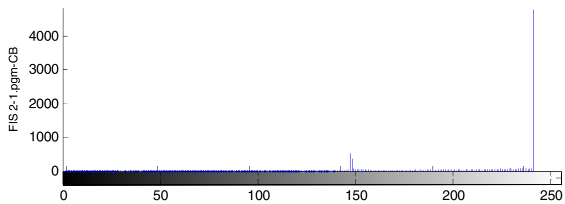
IMAGE: 1.PGM
METHOD: GRADIENT MAGNITUDE

METHOD: FIS 1 (CLEAR BACKGROUND)

METHOD: FIS 2 (CLEAR BACKGROUND)


Table 8.5. Type-2FIS Edges Images Including Only Pixels With Tones Between 150 And 255

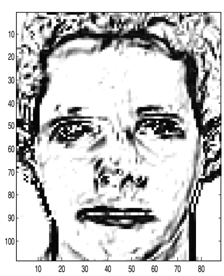
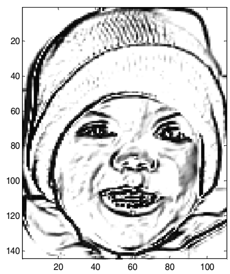
BORDERS IMAGE	DIMENSION (pixels)	PIXELS INCLUDED
	108x88 (9504)	4661 49 %

Table 8.5. (*continued*)

	144x110 (15840)	7077 44.6 %

8.7 Summary

The application of Sobel filters was very useful to define the input vectors for the Type-1 FIS and the Type-2 FIS, although in future works we will try to design Neuro-Fuzzy techniques able to extract image patterns without another data that the original image and to compare the results with traditional techniques of digital signal processing.

Thanks to the histograms of the images it was possible to verify the improvement of results of the FIS Type-1 with respect to the FIS Type-2, since with only the appreciation of the human eye was very difficult to see an objective difference.

The best result was obtained by the Type-2Fuzzy Inference System, because it was possible to clear more than half of the pixels without depreciating the image, which will reduce in drastic form the cost of training in a neural network.

9 Systematic Design of a Stable Type-2 Fuzzy Logic Controller

Stability is one of the more important aspects in the traditional knowledge of Automatic Control. Type-2 Fuzzy Logic is an emerging and promising area for achieving Intelligent Control (in this case, Fuzzy Control). In this chapter we use the Fuzzy Lyapunov Synthesis as proposed by Margaliot to build a Lyapunov Stable Type-1 Fuzzy Logic Control System, and then we make an extension from a Type-1 to a Type-2 Fuzzy Logic Control System, ensuring the stability on the control system and proving the robustness of the corresponding fuzzy controller.

9.1 Introduction

Stability has been one of the central issues concerning fuzzy control since Mamdani's pioneer work (Mamdani and Assilian, 1975). Most of the critical comments to fuzzy control are due to the lack of a general method for its stability analysis.

But as Zadeh often points out, fuzzy control has been accepted by the fact that it is task-oriented control, while conventional control is characterized as setpoint-oriented control, and hence do not need a mathematical analysis of stability. Also, as Sugeno has mentioned, in general, in most industrial applications, the stability of control is not fully guaranteed and the reliability of a control hardware system is considered to be more important than the stability (Sugeno, 1999).

The success of fuzzy control, however, does not imply that we do not need a stability theory for it. Perhaps the main drawback of the lack of stability analysis would be that we cannot take a model-based approach to fuzzy control design. In conventional control theory, a feedback controller can be primarily designed so that a close-loop system becomes stable (Paul and Yang, 1999). This approach of course restricts us to setpoint-oriented control, but stability theory will certainly give us a wider view on the future development of fuzzy control.

Therefore, many researchers have worked to improve the performance of the FLC's and ensure their stability. Li and Gatland in 1995 proposed a more systematic design method for PD and PI-type FLC's. Choi, Kwak and Kim (Choi et al., 2000)

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 109–120, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

present a single-input FLC ensuring stability. Ying in 1994 presented a practical design method for nonlinear fuzzy controllers, and many other researchers have results on the matter of the stability of FLC's, in (Castillo et al., 2005) and (Cázarez et al., 2005) presents an extension of the Margaliot work (Margaliot and G. Langholz, 2000) to built stable type-2 fuzzy logic controllers in Lyapunov sense.

This work is based on Margaliot´s work (Margaliot and Langholtz, 2000), we use the Fuzzy Lyapunov Synthesis to build an Stable Type-2 Fuzzy Logic Controller for a 1 Degree of Freedom (DOF) manipulator robot, first without gravity effect to prove stability, and then with gravity effect to prove the robustness of the controller. The same criteria can be used for any number of DOF manipulator robots, linear or nonlinear, and any kind of plants.

This chapter is organized as follows: In Section 9.2 we present an introductory explanation of type-1 and type-2 FLC's. In Section 9.3 we extend Margaliot's result to build a general rule base for any type (1 or 2) of FLC's. Experimental results are presented in Section 9.4 and the summary is given in Section 9.5.

9.2 Fuzzy Logic Controllers

9.2.1 Type-1 Fuzzy Logic Control

Type-1 FLCs are both intuitive and numerical systems that map crisp inputs to a crisp output. Every FLC is associated with a set of rules with meaningful linguistic interpretations, such as

$$R^l : \text{If } x_1 \text{ is } F_1^l \text{ and } x_2 \text{ is } F_2^l \text{ and } \dots \text{ and } x_n \text{ is } F_n^l \text{ Then } w \text{ is } G^l$$

which can be obtained either from numerical data, or experts familiar with the problem at hand. Based on this kind of statement, actions are combined with rules in an antecedent/consequent format, and then aggregated according to approximate reasoning theory, to produce a nonlinear mapping from input space $U = U_1 x U_2 x ... U_n$ to the output space W, where $F_k^l \subset U_k, k = 1,2,...,n$, are the antecedent type-1 membership functions, and $G^l \subset W$ is the consequent type-1 membership function. The input linguistic variables are denoted by $u_k, k = 1,2,...,n$, and the output linguistic variable is denoted by w.

A Fuzzy Logic System (FLS), as the kernel of a FLC, consist of four basic elements (Fig. 9.1): the type-1 fuzzyfier, the fuzzy rule-base, the inference engine, and the type-1 defuzzyfier. The fuzzy rule-base is a collection of rules in the form of R^l, which are combined in the inference engine, to produce a fuzzy output. The type-1 fuzzyfier maps the crisp input into type-1 fuzzy sets, which are subsequently used as inputs to the inference engine, whereas the type-1 defuzzyfier maps the type-1 fuzzy sets produced by the inference engine into crisp numbers.

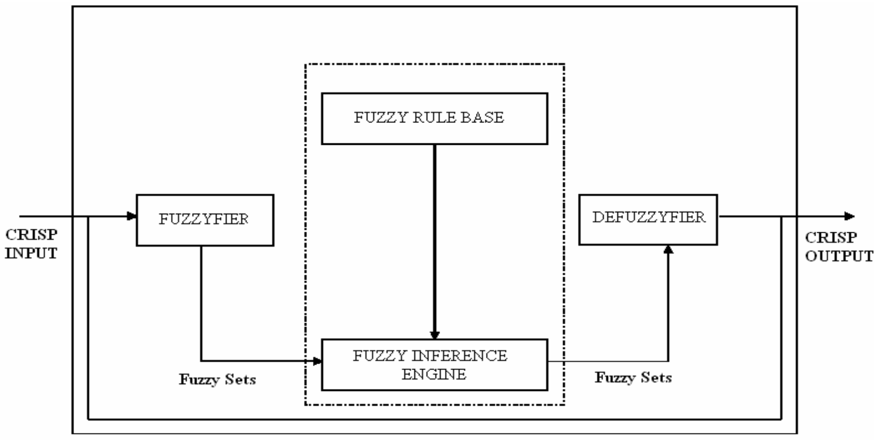


Fig. 9.1. Structure of type-1 fuzzy logic system

Fuzzy sets can be interpreted as membership functions u_X that associate with each element x of the universe of discourse, U, a number $u_X(x)$ in the interval [0,1]:

$$u_X : U \rightarrow [0,1] \tag{9.1}$$

For more detail of Type-1 FLS see (Chen and Pham, 2000).

9.2.2 Type-2 Fuzzy Logic Control

As with the type-1 fuzzy set, the concept of type-2 fuzzy set was introduced by Zadeh as an extension of the concept of an ordinary fuzzy set (Zadeh, 1975).

A FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain. On the other hand, type-2 FLSs, are very useful in circumstances where it is difficult to determine an exact, and measurement uncertainties (Mendel, 2000).

It is known that type-2 fuzzy set let us to model and to minimize the effects of uncertainties in rule-based FLS. Unfortunately, type-2 fuzzy sets are more difficult to use and understand that type-1 fuzzy sets; hence, their use is not widespread yet.

Similar to a type-1 FLS, a type-2 FLS includes type-2 fuzzyfier, rule-base, inference engine and substitutes the defuzzifier by the output processor. The output processor includes a type-reducer and a type-2 defuzzyfier; it generates a type-1 fuzzy set output (from the type reducer) or a crisp number (from the defuzzyfier). A type-2 FLS is again characterized by IF-THEN rules, but its antecedent of consequent sets are now type-2. Type-2 FLSs, can be used when the circumstances are too uncertain to determine exact membership grades. A model of a type-2 FLS is shown in Fig. 9.2.

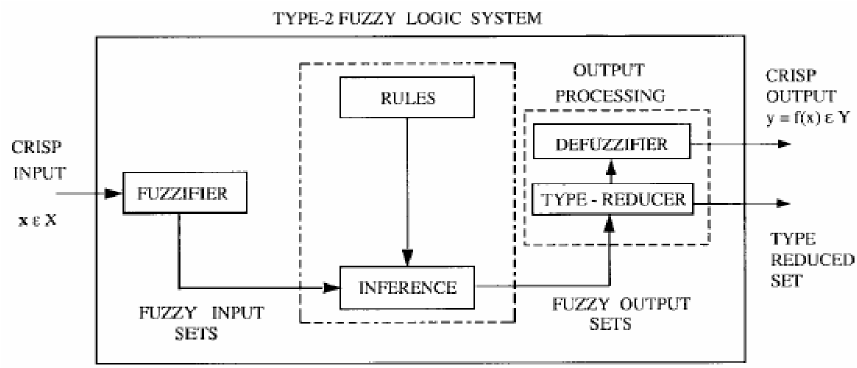


Fig. 9.2. Structure of type-2 fuzzy logic system

In the case of the implementation of type-2 FLCs, we have the same characteristics as in type-1 FLC, but we now use type-2 fuzzy sets as membership functions for the inputs and for the outputs. Fig. 9.3 shows the structure of a control loop with a FLC.

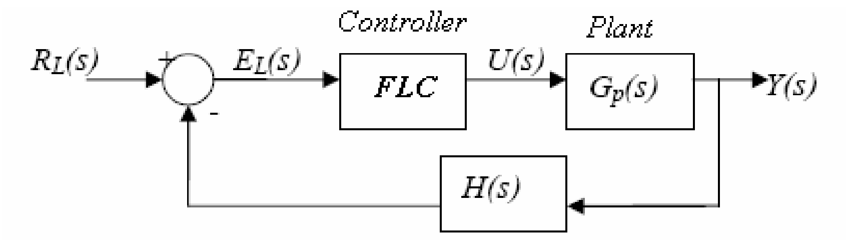


Fig. 9.3. Fuzzy control loop

9.3 Systematic and Design of Stable Fuzzy Controllers

For our description we consider the problem of designing a stabilizing controller for a 1DOF manipulator robot system depicted in Fig.9.4. The state-variables are $x_1 = \theta$ - the robot arm angle, and $x_2 = \dot{\theta}$ - its angular velocity. The system's actual dynamical equation, which we will assume unknown, is as shown in equation (9.2) (Paul and Yang, 1999):

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{9.2}$$

To apply the fuzzy Lyapunov synthesis method, we assume that the exact equations are unknown and that we have only the following partial knowledge about the plant (see Fig. 9.4):

1. The system may have really two degrees of freedom θ and $\dot{\theta}$, referred to as x_1 and x_2, respectively. Hence, $\dot{x}_1 = x_2$.

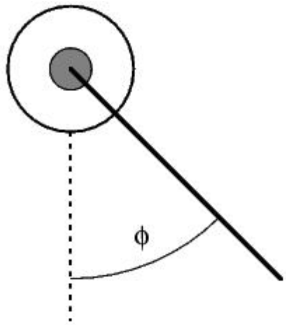


Fig. 9.4. 1DOF Manipulator robot

2. $\dot{x}_2$ is proportional to u, that is, when u increases (decreases) $\dot{x}_2$ increases (decreases).

To facilitate our control design we are going to suppose no gravity effect in our model, see (equation 9.3).

$$ml^2\ddot{q} = \tau \tag{9.3}$$

Our objective is to design the rule-base of a fuzzy controller that will carry the robot arm to a desired position $x_1 = \theta d$. We choose (9.4) as our Lyapunov function candidate. Clearly, V is positive-definite.

$$V(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2) \tag{9.4}$$

Differentiating V, we have (9.5),

$$\dot{V} = x_1\dot{x}_1 + x_2\dot{x}_2 = x_1x_2 + x_2\dot{x}_2 \tag{9.5}$$

Hence, we require:

$$x_1x_2 + x_2\dot{x}_2 < 0 \tag{9.6}$$

We can now derive sufficient conditions so that condition (9.6) holds: If x_1 and x_2 have opposite signs, then $x_1x_2 < 0$ and (9.6) will hold if $\dot{x}_2 = 0$; if x_1 and x_2 are both positive, then (9.6) will hold if $\dot{x}_2 < -x_1$; and if x_1 and x_2 are both negative, then (9.6) will hold if $\dot{x}_2 > -x_1$.

We can translate these conditions into the following fuzzy rules:

- If x_1 is *positive* and x_2 is *positive* then $\dot{x}_2$ must be *negative big*
- If x_1 is *negative* and x_2 is *negative* then $\dot{x}_2$ must be *positive big*
- If x_1 is *positive* and x_2 is *negative* then $\dot{x}_2$ must be *zero*
- If x_1 is *negative* and x_2 is *positive* then $\dot{x}_2$ must be *zero*

However, using our knowledge that $\dot{x}_2$ is proportional to u, we can replace each $\dot{x}_2$ with u to obtain the fuzzy rule-base for the stabilizing controller:

- If x_1 is *positive* and x_2 is *positive* Then u must be *negative big*
- If x_1 is *negative* and x_2 is *negative* Then u must be *positive big*
- If x_1 is *positive* and x_2 is *negative* Then u must be *zero*
- If x_1 is *negative* and x_2 is *positive* Then u must be *zero*

It is interesting to note that the fuzzy partitions for x_1, x_2, and u follow elegantly from expression (9.5). Because $\dot{V} = x_2(x_1 + \dot{x}_2)$, and since we require that $\dot{V}$ be negative, it is natural to examine the signs of x_1 and x_2; hence, the obvious fuzzy partition is *positive*, *negative*. The partition for $\dot{x}_2$, namely *negative big*, *zero*, *positive big* is obtained similarly when we plug the linguistic values *positive*, *negative* for x_1 and x_2 in (9.5). To ensure that $\dot{x}_2 < -x_1$ $(\dot{x}_2 > -x_1)$ is satisfied even though we do not know x_1's exact magnitude, only that it is *positive* (*negative*), we must set $\dot{x}_2$ to *negative big* (*positive big*). Obviously, it is also possible to start with a given, pre-defined, partition for the variables and then plug each value in the expression for $\dot{V}$ to find the rules. Nevertheless, regardless of what comes first, we see that fuzzy Lyapunov synthesis transforms classical Lyapunov synthesis from the world of exact mathematical quantities to the world of computing with words (Zadeh, 1996).

To complete the controllers design, we must model the linguistic terms in the rule-base using fuzzy membership functions and determine an inference method. Following (Wang, 1997), we characterize the linguistic terms *positive*, *negative*, *negative big*, *zero* and *positive big* by the type-1 membership functions shown in Fig. 9.5 for a Type-1 Fuzzy Logic Controller, and by the type-2 membership functions shown in Fig. 9.6 for a Type-2 Fuzzy Logic Controller. Note that the type-2 membership functions are extended type-1 membership functions.

To this end, we had systematically developed a FLC rule-base that follows the Lyapunov Stability criterion. In Section 9.4 we present some experimental results using our fuzzy rule-base to build a Type-2 Fuzzy Logic Controller.

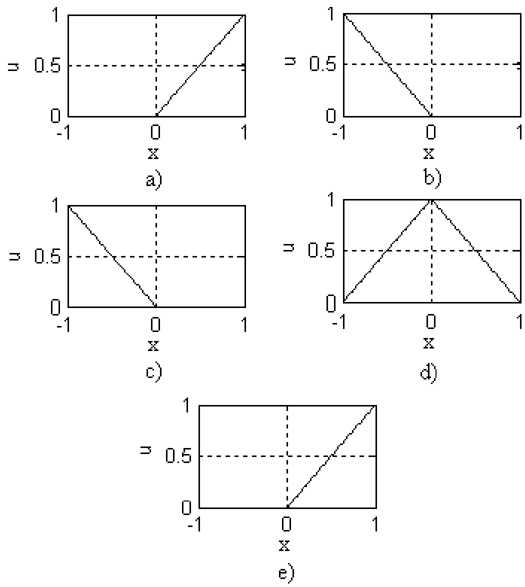


Fig. 9.5. Set of type-1 membership functions: a) positive, b)negative, c) negative big, d) zero and e) positive big

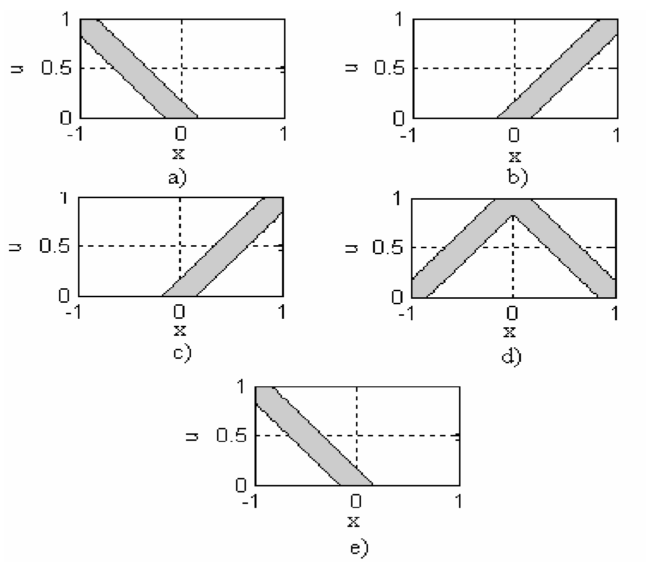


Fig. 9.6. Set of type-2 membership functions: a)negative, b) positive, c) positive big, d) zero and e) negative big

9.4 Experimental Results

In Section 9.3 we had systematically developed a stable FLC rule-base, and now we are going to show some experimental results using our stable rule-base to build a Type-2 FLC. The plant description used in the experiments is the same shown in Section 9.3.

Our experiments were done with Type-1 Fuzzy Sets and Interval Type-2 Fuzzy Sets. In the Type-2 Fuzzy Sets the membership grade of every domain point is a crisp set whose domain is some interval contained in [0,1] (Mendel, 2000). On Fig. 9.6 we show some Interval Type-2 Fuzzy Sets, and for each fuzzy set, the grey area is known as the Footprint of Uncertainty (FOU) (Mendel, 2000), and this is bounded by an upper and a lower membership function as shown in Fig. 9.7.

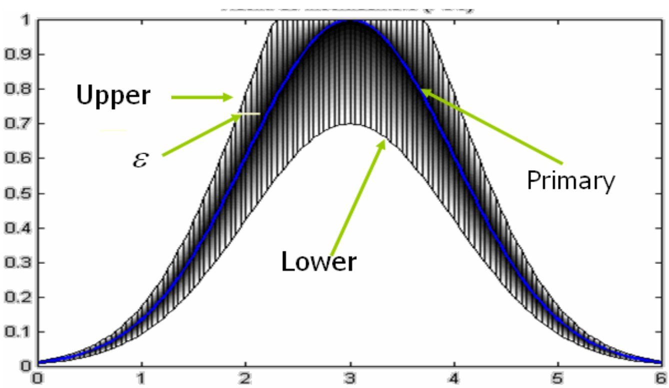


Fig. 9.7. Interval Type-2 Fuzzy Set

In our experiments we increase and decrease the value of ε to the left and to the right side having a εL and a εR values respectively to determine how much the FOU can be extended or perturbed without losing stability in the FLC.

We did make simulations with initial conditions of θ having values in the whole circumference [0, 2π], and the desired angle θd having values in the same range. The initial conditions considered in the experiments shown in this paper are an angle $\theta = 0 rad$ and $\theta_d = 0.1 rad$.

In Fig. 9.8 we show a simulation of the plant made with a Type-1 FLC, as can be seen, the plant has been regulated in around 8 seconds, and in Fig. 9.9 we show the graph of equation (9.5) which is always negative defined and consequently the system is stable.

Figure 9.10 shows the simulation results of the plant made with the Type-2 FLC increasing and decreasing ε in the range of [0,1], and as can be seen the plant has been regulated in around 10 seconds, and the graph of (9.5), which is depicted in Fig. 9.11 is always negative defined and consequently the system is stable. As we can seen, the time response is increasing when the value of ε is increasing.

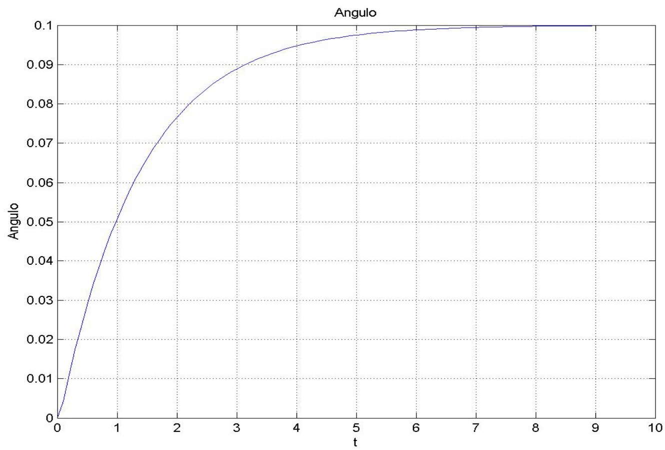


Fig. 9.8. Response for the Type-1 FLC

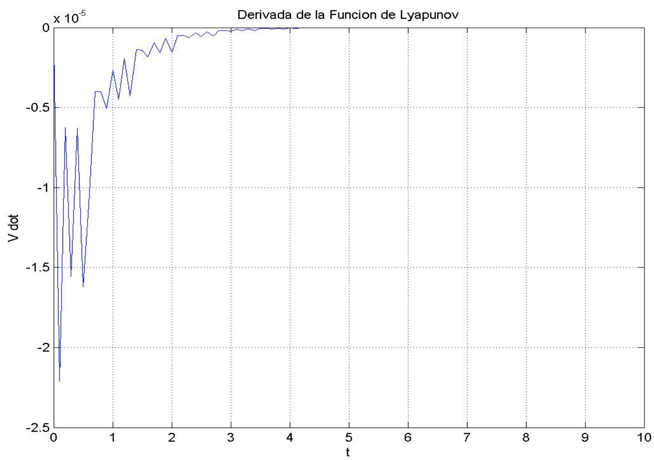


Fig. 9.9. $\dot{V}$ for the Type-1 FLC

With the variation of ε in the definition of the FOU, the control surface changes proportional to the change of ε, for this reason, the value of *u* for $\varepsilon \geq 1$ is practically zero, and the plant does not have physical response. To test the robustness of the built Fuzzy Controller, now we are going to use the same controller designed in Section 9.3, but at this time, we are going to use it to control equation (9.2) considering the gravity effect as shown in equation (9.7).

$$ml^2\ddot{q} + gml\cos q = \tau \tag{9.7}$$

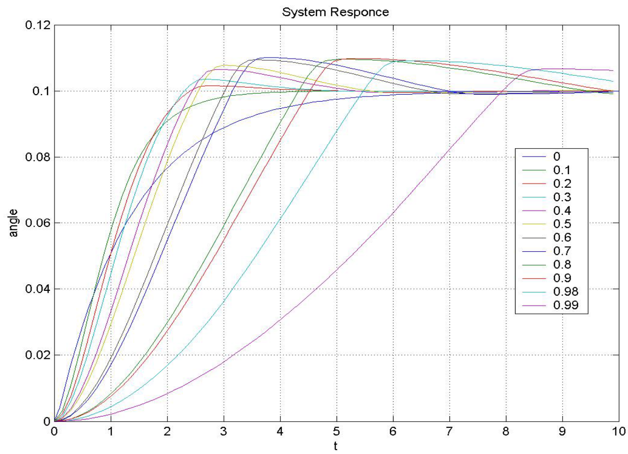


Fig. 9.10. Response for the Type-2 FLC ($\varepsilon \rightarrow [0,1)$)

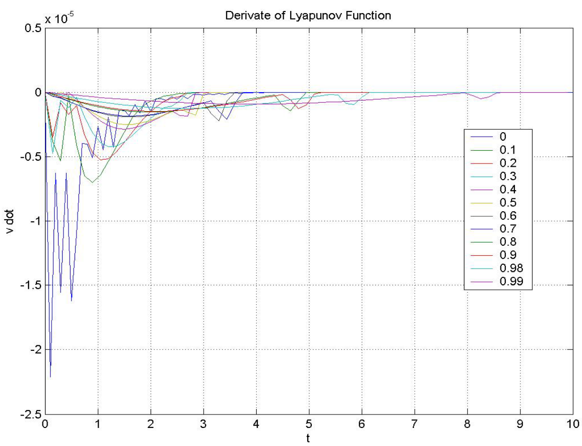


Fig. 9.11. $\dot{V}$ for the Type-2 FLC ($\varepsilon \rightarrow [0,1]$)

In Figure 9.12 we can see a simulation of the plant obtained with a Type-1 FLC, and as can be seen, the plant has been regulated in approximately 8 seconds and Figure 9.13 shows the graph of (9.5) which is always negative defined and consequently the system is stable.

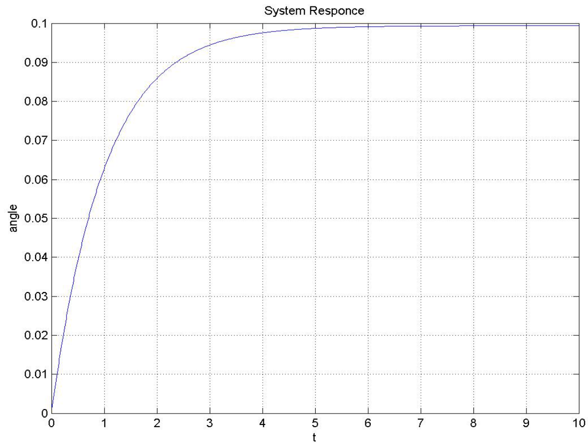


Fig. 9.12. Response for the Type-1 FLC

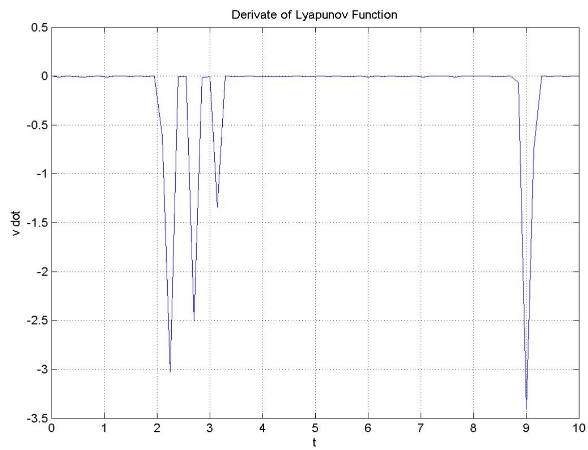


Fig. 9.13. $\dot{V}$ for the Type-1 FLC

Figure 9.14 shows the simulation results of the plant obtained with the Type-2 FLC with increasing and decreasing ε values in the range of [0,1], and the graph of (9.5) depicted at Fig. 9.15 is always negative defined and consequently the system is stable. As we can seen, if we use an adaptive gain like in (Castillo et al., 2005) all the cases of ε can be regulated around 8 seconds.

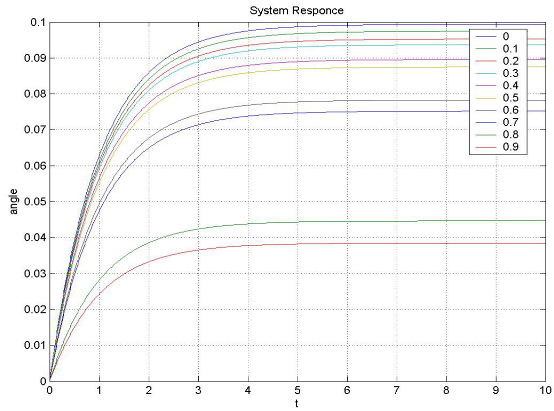


Fig. 9.14. Response for the Type-2 FLC ($\varepsilon \to [0,1]$)

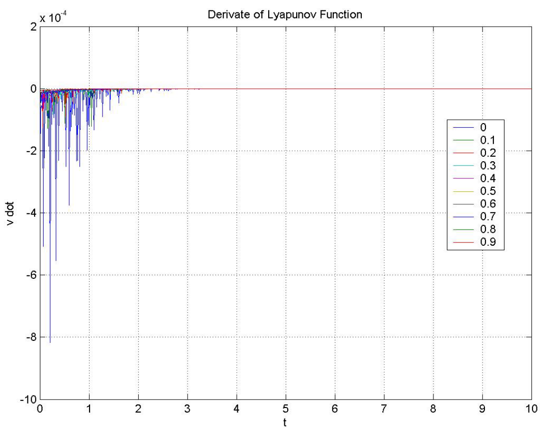


Fig. 9.15. $\dot{V}$ for the Type-2 FLC ($\varepsilon \to [0,1]$)

9.5 Summary

Margaliot's approach for the design of FLC's is now proved to be valid for both, Type-1 and Type-2 Fuzzy Logic Controllers. In the case of Type-2 FLC's membership functions, we can perturb or change the definition domain of the FOU without losing of stability of the controller; in the case described in this chapter, like in (Castillo et al.,2005) we have to use an adaptive gain to regulate the plant in a desired time. For our example of the 1DOF manipulator robot, stability holds when extending the FOU on the domain [0,1), and we also have shown that a FLC designed following the Fuzzy Lyapunov Synthesis is stable and robust.

10 Experimental Study of Intelligent Controllers Under Uncertainty Using Type-1 and Type-2 Fuzzy Logic

Uncertainty is an inherent part in controllers used for real-world applications. The use of new methods for handling incomplete information is of fundamental importance in engineering applications. This chapter deals with the design of controllers using type-2 fuzzy logic for minimizing the effects of uncertainty produced by the instrumentation elements. We simulated type-1 and type-2 fuzzy logic controllers to perform a comparative analysis of the systems' response, in the presence of uncertainty.

10.1 Introduction

Uncertainty affects decision-making and appears in a number of different forms. The concept of information is fully connected with the concept of uncertainty. The most fundamental aspect of this connection is that the uncertainty involved in any problem-solving situation is a result of some information deficiency, which may be incomplete, imprecise, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way (Klir and Yuan, 1995). The general framework of fuzzy reasoning allows handling much of this uncertainty, fuzzy systems employ type-1 fuzzy sets, which represents uncertainty by numbers in the range [0, 1]. However, when something is uncertain, like a measurement, it is difficult to determine its exact value, and of course type-1 fuzzy sets make more sense than using crisp sets (Zadeh, 1975). However, it is not reasonable to use an accurate membership function for something uncertain, so in this case what we need is another type of fuzzy sets, those, which are able to handle these uncertainties, the so called type-2 fuzzy sets (Mendel, 2000). So, the amount of uncertainty in a system can be reduced by using type-2 fuzzy logic because it offers better capabilities to handle linguistic uncertainties by modeling vagueness and unreliability of information (Karnik and Mendel, 2001).

Recently, we have seen the use of type-2 fuzzy sets in fuzzy logic systems to deal with uncertain information (Mendel, 1998). So we can find some papers emphasizing on the implementation of a type-2 Fuzzy Logic System (FLS) (Karnik and Mendel, 1999); in others, it is explained how type-2 fuzzy sets let us model and minimize the

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 121–132, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

effects of uncertainties in rule-base FLSs (Mendel and John, 2002). Some research works are devoted to solve real world applications in different areas, for example, in signal processing type-2 fuzzy logic is applied in prediction in Mackey-Glass chaotic time-series with uniform noise presence (Mendel, 2000). In medicine, an expert system was developed for solving the problem of Umbilical Acid-Base (UAB) assessment (Ozen and Garibaldi, 2003). In industry, type-2 fuzzy logic and neural networks was used in the control of non-linear dynamic plants (Hagras, 2004) (Melin and Castillo, 2004).

This chapter deals with the advantages of using type-2 fuzzy sets in the implementation of a Fuzzy Logic Controller (FLC), for a real system. It is a fact, that in the control of real systems, the instrumentation elements (instrumentation amplifier, sensors, digital to analog, analog to digital converters, etc.) introduce some sort of unpredictable values in the information that has been collected (Castillo and Melin, 2001). So, the controllers designed under idealized conditions tend to behave in an inappropriate manner. Since, uncertainty is inherent in the design of controllers for real world applications, we are presenting how to deal with it using type-2 FLC to diminish the effects of imprecise information. We are supporting this statement with experimental results, qualitative observations, and quantitative measures of errors. For quantifying the errors, we utilized three widely used performance criteria, these are: Integral of Square Error (ISE), Integral of the Absolute value of the Error (IAE), and Integral of the Time multiplied by the Absolute value of the Error (ITAE) (Sepulveda et al., 2007).

This chapter is organized as follows: section 10.2 presents an introductory explanation of type-1 and type-2 FLCs and the performance criteria for evaluating the transient and steady state closed-loop response in a computer control system. In section 10.3, we are showing details of the implementation of the feedback control system used in this work, we are presenting some experimental results and a performance comparison between type-1 and type-2 fuzzy logic controllers.

10.2 Fuzzy Controllers

In the 40's and 50's, many researchers proved that many dynamic systems can be mathematically modeled using differential equations. These previous works represent the foundations of the Control theory which, in addition with the Transform theory, provided an extremely powerful means of analyzing and designing control systems. These theories were being developed until the 70's, when the area was called System theory to indicate its definitiveness (Mamdani, 1993). Its principles have been used to control a very big amount of systems taking mathematics as the main tool to do it during many years. Unfortunately, in too many cases this approach could not be sustained because many systems have unknown parameters or highly complex and nonlinear characteristics that make them not to be amenable to the full force of mathematical analysis as dictated by the Control theory.

Soft computing techniques have become a research topic, which is applied in the design of controllers (Jang et al., 1997). These techniques have tried to avoid the above-mentioned drawbacks, and they allow us to obtain efficient controllers, which utilize the human experience in a more related form than the conventional mathematical approach

(Zadeh, 1971). In the cases in which a mathematical representation of the controlled systems cannot be obtained, the process operator should be able to express the relationships existing in them, that is, the process behavior.

A FLS, described completely in terms of type-1 fuzzy sets is called a type-1 fuzzy logic system (type-1 FLS). It is composed by a knowledge base that comprises the information given by the process operator in form of linguistic control rules, a fuzzification interface, who has the effect of transforming crisp data into fuzzy sets, an inference system, that uses them in conjunction with the knowledge base to make inference by means of a reasoning method, and a defuzzification interface, which translates the fuzzy control action so obtained to a real control action using a defuzzification method.

In this chapter, the implementation of the fuzzy controller in terms of type-1 fuzzy sets, has two input variables such as the error $e(t)$, the difference between the reference signal and the output of the process, as well as the error variation $\Delta e(t)$,

$$e(t) = r(t) - y(t) \tag{10.1}$$

$$\Delta e(t) = e(t) - e(t-1) \tag{10.2}$$

so the control law can be represented as in Fig. 10.1.

A FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain (Mendel and Mouzouris, 1999). On the other hand, type-2 FLSs, are very useful in circumstances where it is difficult to determine an exact certainties, and measurement uncertainties (Mendel, 2000).

It is known that type-2 fuzzy sets let us to model and to minimize the effects of uncertainties in rule-based FLS. Unfortunately, type-2 fuzzy sets are more difficult to use and understand than type-1 fuzzy sets; hence, their use is not widespread yet. In (Sepulveda et al., 2007) were mentioned at least four sources of uncertainties in type-1 FLSs:

1. The meanings of the words that are used in the antecedents and consequents of rules can be uncertain (words mean different things to different people).
2. Consequents may have histogram of values associated with them, especially when knowledge is extracted from a group of experts who do not all agree.
3. Measurements that activate a type-1 FLS may be noisy and therefore uncertain.
4. The data used to tune the parameters of a type-1 FLS may also be noisy.

All of these uncertainties translate into uncertainties about fuzzy set membership functions. Type-1 fuzzy sets are not able to directly model such uncertainties because their membership functions are totally crisp. On the other hand, type-2 fuzzy sets are able to model such uncertainties because their membership functions are themselves fuzzy. A type-2 membership grade can be any subset in [0,1], the primary membership, and corresponding to each primary membership, there is a secondary membership (which can also be in [0,1]) that defines the possibilities for the primary

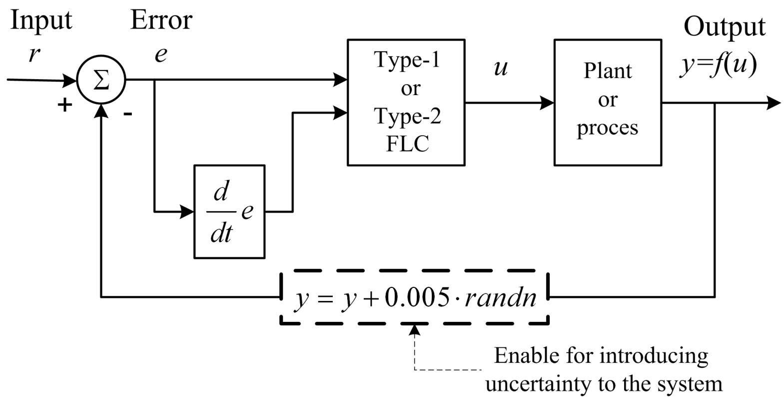


Fig. 10.1. Block diagram of the fuzzy control

membership. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity.

Similar to a type-1 FLS, a type-2 FLS includes fuzzifier, rule base, fuzzy inference engine, and output processor. The output processor includes type-reducer and defuzzifier; it generates a type-1 fuzzy set output (from the type-reducer) or a crisp number (from the defuzzifier). A type-2 FLS is again characterized by IF-THEN rules, but its antecedent or consequent sets are now type-2. Type-2 FLSs, can be used when the circumstances are too uncertain to determine exact membership grades such as when training data is corrupted by noise. In our case, we are simulating that the instrumentation elements (instrumentation amplifier, sensors, digital to analog, analog to digital converters, etc.) are introducing some sort of unpredictable values in the collected information.

In the case of the implementation of the type-2 FLC, we have the same characteristics as in type-1 FLC, but we used type-2 fuzzy sets as membership functions for the inputs and for the output.

For evaluating the transient closed-loop response of a computer control system we can use the same criteria that normally are used for adjusting constants in PID (Proportional Integral Derivative) controllers. These are (Deshpande and Ash, 1988):

1. Integral of Square Error (ISE).

$$\mathrm{ISE} = \int_0^\infty [e]^2 \, dt \tag{10.3}$$

2. Integral of the Absolute value of the Error (IAE).

$$\mathrm{IAE} = \int_0^\infty | e | \, dt \tag{10.4}$$

3. Integral of the Time multiplied by the Absolute value of the Error (ITAE).

$$\mathrm{ITAE} = \int_0^\infty t \, | e | \, dt \tag{10.5}$$

The selection of the criteria depends on the type of response desired, the errors will contribute different for each criterion, so we have that large errors will increase the value of ISE more heavily than to IAE. ISE will favor responses with smaller overshoot for load changes, but ISE will give longer settling time. In ITAE, time appears as a factor, and therefore, ITAE will penalize heavily on errors that occur late in time, but virtually ignores errors that occurs early in time. Designing using ITAE will give us the shortest settling time, but it will produce the largest overshoot among the three criteria considered. Designing considering IAE will give us an intermediate result, in this case, the settling time will not be so large than using ISE nor so small than using ITAE, and the same applies for the overshoot response. The selection of a particular criterion is depending on the type of desired response.

10.3 Experimental Results

We are showing in Fig. 10.1, the feedback control system that was used for achieving the results of this paper. It was implemented in Matlab where the controller was designed to follow the input as closely as possible. The plant was modeled using equation (10.6)

$$y(i)=0.2\cdot y(i-3)\cdot 0.7y(i-2)+0.9\cdot y(i-1)+0.005u(i-1)+0.5\cdot u(i-2) \qquad (10.6)$$

The controller's output was applied directly to the plant's input. Since we are interested in comparing the performance between type-1 and type-2 FLC system, we tested the controller in two ways:

1. One is considering the system as ideal, that is, we did not introduce in the modules of the control system any source of uncertainty. See experiments 1, and 2.
2. The other one is simulating the effects of uncertain modules (subsystems) response introducing some uncertainty. See experiments 3, and 4.

For both cases, as is shown in Fig. 10.1, the system's output is directly connected to the summing junction, but in the second case, the uncertainty was simulated introducing random noise with normal distribution (the dashed square in Fig. 10.1). We added noise to the system's output $y(i)$ using equation (10.7), which in turn was introduced to the summing junction of the controller system.

$$y(i)=y(i)+0.05\cdot randn \qquad (10.7)$$

We tested the system using as input, a unit step sequence free of noise, $r(i)$. For evaluating the system's response and compare between type 1 and type 2 fuzzy controllers, we used the performance criteria ISE, IAE, and ITAE. In table 10.1, we summarized the values obtained for each criterion considering 400 units of time. For calculating ITAE we considered a sampling time $T_s = 0.1$ sec.

For Experiments 1, 2, 3, and 4 the reference input r is stable and noisy free. In experiments 3 and 4, although the reference appears clean, the feedback at the summing junction is noisy since we introduced deliberately noise for simulating the overall existing uncertainty in the system, in consequence, the controller's inputs e (error), and $\frac{\Delta}{\Delta t}e$ contains uncertainty data.

For each input of the type-1 FLC, we defined three type-1 fuzzy Gaussian membership functions: negative, zero, positive. The universe of discourse for these membership functions is in the range [-10 10]; their mean is -10, 0 and 10 respectively, and their standard deviation are 9, 2 and 9 respectively.

For the output, we have five type-1 fuzzy Gaussian membership functions: NG, N, Z, P and PG. They are on the interval [-10 10], their means are -10, -4.5, 0, 4, and 10 respectively; and their standard deviations are 4.5, 4, 4.5, 4 and 4.5 respectively.

In the type-2 FLC, for each input we defined three type-2 fuzzy Gaussian membership functions: negative, zero, positive. In this case the fuzzy membership functions have uncertain mean and fixed standard deviation on the interval [-10 10]. For the upper membership functions we have -10.5, -1, and 9.5 uncertain means; for the lower membership functions we have -9.5, 1, and 10.5 uncertain means respectively; for the fixed standard deviations 9, 2 and 9 respectively.

For computing the output we have five type-2 fuzzy Gaussian membership functions with uncertain mean and fixed standard deviations: NG, N, Z, P and PG, on the interval [-10 10]. For the upper membership functions we have -10.25, -4.75, -0.25, 3.75 and 9.75 uncertain means; for the lower membership functions we have -9.75, -4.25, 0.25, 4.25 and 10.25 uncertain means respectively. The fixed standard deviations: 4.5, 4, 4.5, 4 and 4.5 respectively.

For the type-2 FLC, the inputs and the output have interval type-2 membership functions. In all of the experiments, we used a dash-dot line for illustrating the system's response and behavior of type-1 FLC, in the same sense, we used continuous line for type-2 FLC. The reference input r is shown with a dot line.

Experiment 1. Ideal system using a type-1 FLC.
In this experiment, we did not add uncertainty data to the system, the system response is illustrated in Figure 10.2. Note that the settling time is in about 140 units of time; i.e., the system trends to stabilize with time and the output will follow accurately the input. In Table 10.1, we listed the obtained values of ISE, IAE, and ITAE for this experiment. We are showing in Fig. 10.3, 10.4 and 10.5 the ISE, IAE, and ITAE behavior of this experiment.

Experiment 2. Ideal system using a type-2 FLC.
Here, we used the same test conditions of Experiment 1, but in this case, we implemented the controller's algorithm with type-2 fuzzy logic, its output sequence is illustrated in Fig. 10.2, and the corresponding performance criteria are listed in Table 10.1. By visual inspection, we can observe that the output system response of the Experiment 1, and this one, are very similar, they are almost overlapped.

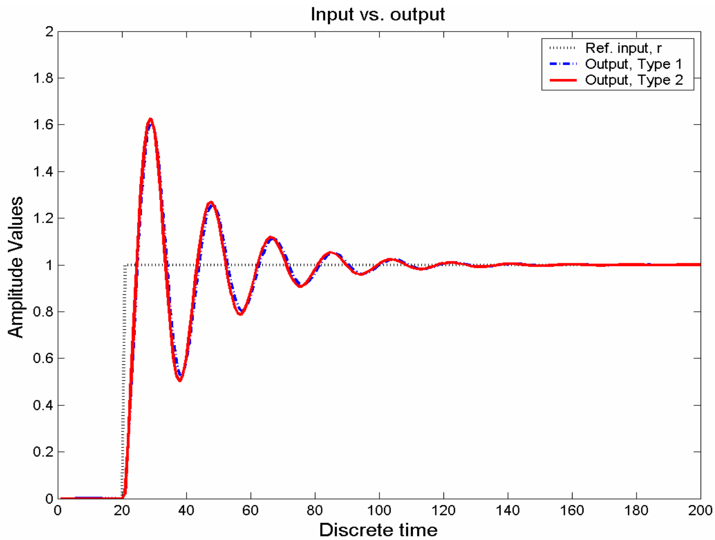


Fig. 10.2. This graphic shows the system's response to a unit step sequence. The input reference r is shown with pointed line, for type 1 the systems's output $y(i)$ is shown with dash dot line; and for type-2, the system's output $y(i)$ with continuous line. Note, that both responses are very similar, although, in this case the lower errors were obtained with type-1 FLC.

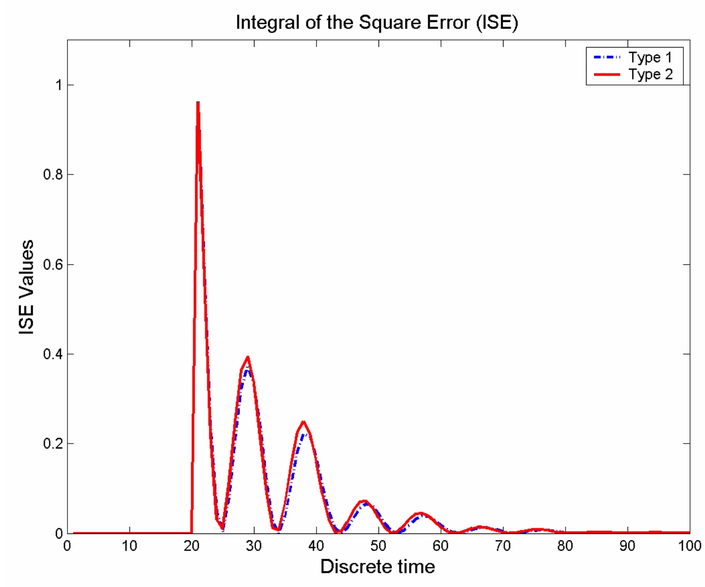


Fig. 10.3. In uncertainty absence, the ISE values are very similar for type-1 and type-2 FLCs

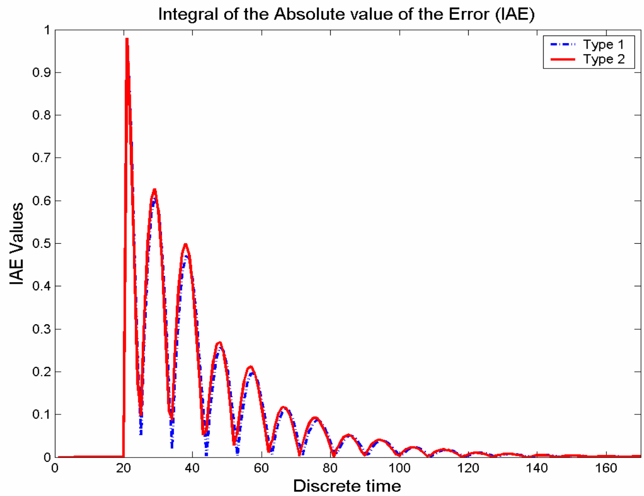


Fig. 10.4. In uncertainty absence, the IAE values obtained at the plant's output are very similar for type-1 and type-2 FLCs, here is more evident that a type-1 FLC works a little better than in Fig. 10.3

Using the performance criteria we can get a quantitative comparison, where we can observe small differences favoring Experiment 1, i.e., the results obtained using a type-1 FLC.

We can observe in Fig. 10.3, 10.4, and 10.5 that using a type-1 FLC we got the lower errors.

Experiment 3. System with uncertainty using a type-1 FLC.
In this case, we simulated using equation (7), the effects of uncertainty introduced to the system by transducers, amplifiers, and any other element that in real world applications affects expected values. We are showing in Fig. 6, the system's response output. In Fig. 10.7, 10.8, and 10.9 are plotted the performance criteria ISE, IAE, ITAE.

Experiment 4. System with uncertainty using a type-2 FLC. In this experiment, we introduced uncertainty in the system, in the same way as in Experiment 3. In this case, we used a type-2 FLC and we improved those results obtained with a type-1 FLC (Experiment 3).

We can easily appreciate in Fig. 10.6, that the lower overshoot and the best settling times were reached using a type-2 FLC.

Using Fig. 10.7 and 10.8, we can see that with a type-2 FLC the overshoot error decreases very quickly and remains lower than using a type-1 FLC. In Fig. 10.9, we can observe that through the time the lower errors are obtained using a type-2 FLC.

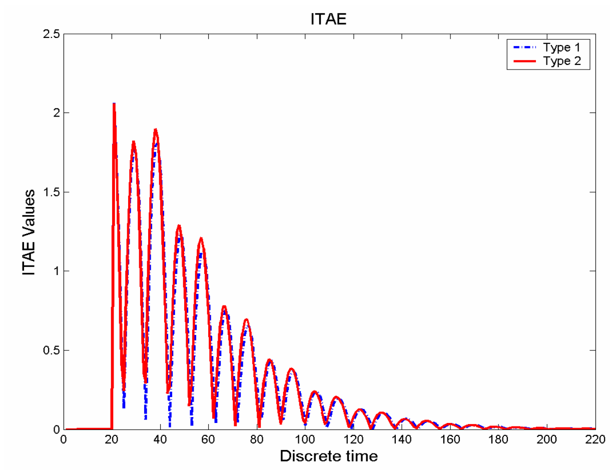


Fig. 10.5. In uncertainty absence, the ITAE values obtained at the plant's output are very similar for type-1 and type-2 FLCs, in accordance with Figure 10.13, it is evident a type-1 FLC works a little better

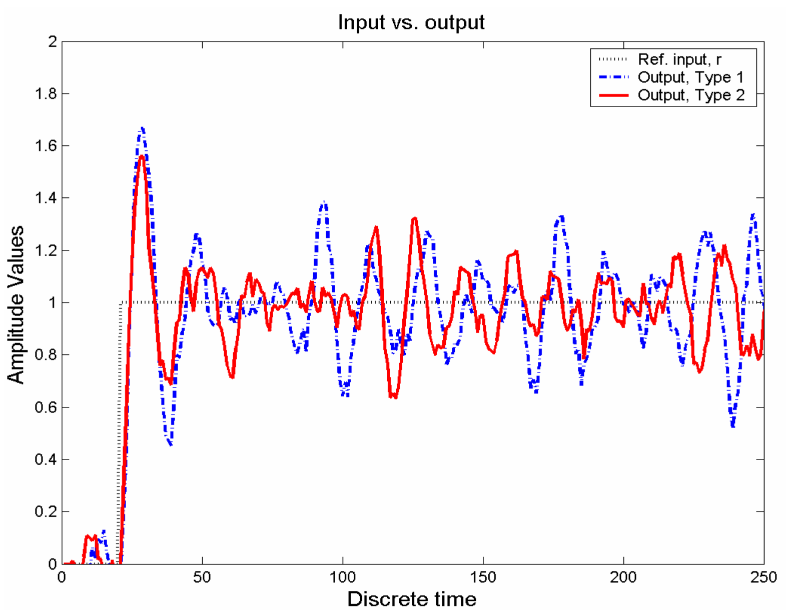


Fig. 10.6. This graphic was obtained with uncertainty presence; compare the system's outputs produced by type-1 and type-2 FLCs. Note that quite the opposite of Figure 10.2, a type-2 FLC works much better than a type-1 FLC when the system has uncertainty. The overshoot error is lower for a type-2 FLC.

Table 10.1. comparison of performance criteria for type-1 and type-2 fuzzy logic controllers. Values obtained after 400 samples.

Performance Criteria	Type-1 FLC		Type-2 FLC	
	Ideal System	Syst. with uncertainty	Ideal System	Syst. with uncertainty
ISE	5.2569	15.1143	5.4479	9.5516
IAE	13.8092	57.9542	14.204	45.4106
ITAE	59.9589	1111.2	61.636	877.5299

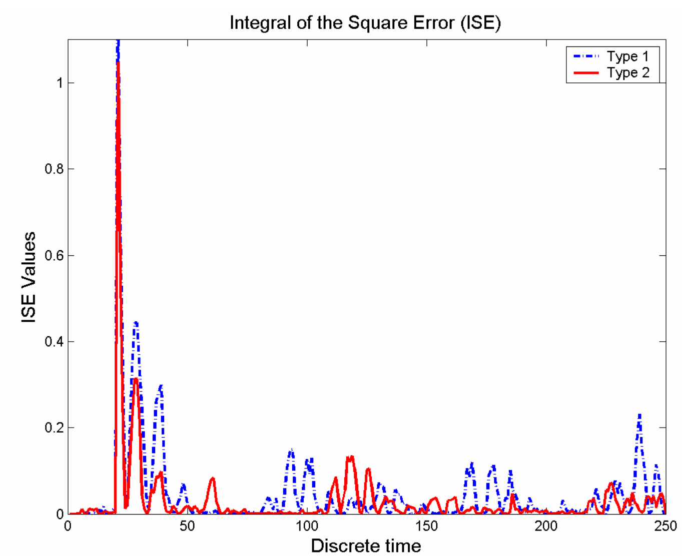


Fig. 10.7. Here we can see that a type-2 FLC produces lower overshoot errors, quantitatively the ISE overall error of using type-2 is 9.5516 against 15.1143 of the overall error produced by the type-1 FLC

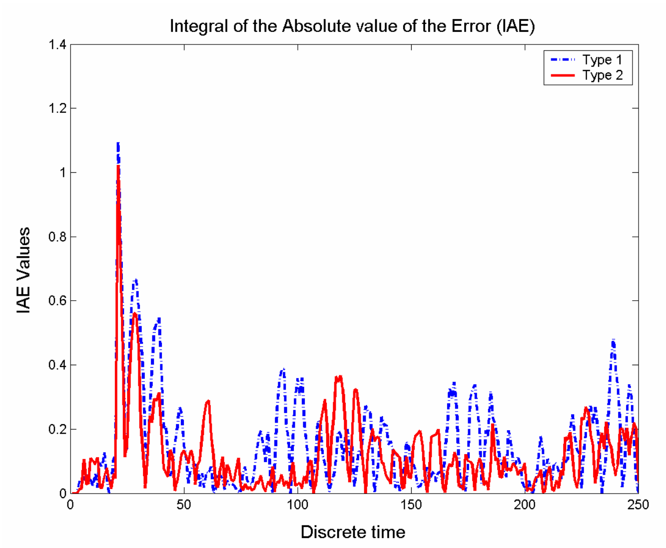


Fig. 10.8. In accordance with Fig. 10.6, IAE confirms that we obtained the best system response using a type-2 FLC with uncertainty presence. Moreover, the error of the settling time and steady state is lower using a type-2 FLC.

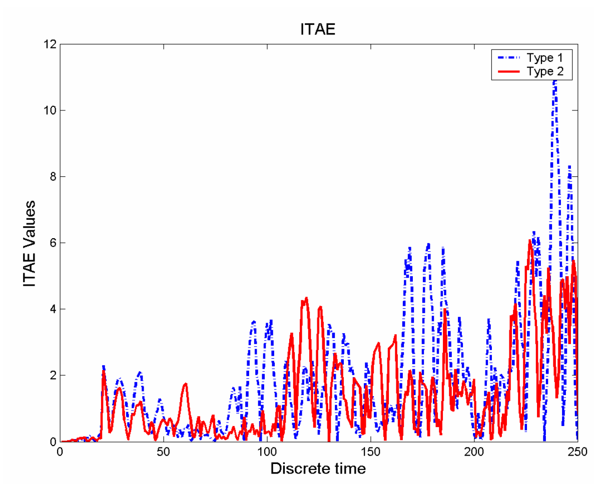


Fig. 10.9. Here we can see that the steady state error of the system produced by a type-2 FLC is lower than the error produced by a type-1 FLC with uncertainty present. ITAE will punish heavily all those errors produced with time.

10.4 Summary

We observed and quantified using performance criteria such as ISE, IAE, and ITAE that in systems without uncertainties (ideal systems) is a better choice to select a type-1 FLC since it works a little better than a type-2 FLC, and it is easier to implement it. It is known that type-1 FLC can handle nonlinearities, and uncertainties up to some extent.

Unfortunately, real systems are inherently noisy and nonlinear, since any element in the system contributes with deviations of the expected measures because of thermal noise, electromagnetic interference, etc., moreover, they add nonlinearities from element to element in the system.

For real systems, systems with uncertainty, we observed and quantify that the lower overshoot errors and the best settling times were obtained using a type-2 FLC.

We are concluding that using a type-2 FLC in real world applications can be a better choice since the amount of uncertainty in real systems most of time is difficult to estimate.

11 Evolutionary Optimization of Interval Type-2 Membership Functions Using the Human Evolutionary Model

Uncertainty is an inherent part in controllers used for real-world applications. The use of new methods for handling incomplete information is of fundamental importance in engineering applications. We simulated the effects of uncertainty produced by the instrumentation elements in type-1 and type-2 fuzzy logic controllers to perform a comparative analysis of the systems' response, in the presence of uncertainty. We are presenting an innovative idea to optimize interval type-2 membership functions using an average of two type-1 systems with the Human Evolutionary Model, we are showing comparative results of the optimized proposed method. We found that the optimized membership functions for the inputs of a type-2 system increases the performance of the system for high noise levels.

11.1 Introduction

Uncertainty affects decision-making and appears in a number of different forms. The concept of information is fully connected with the concept of uncertainty. The most fundamental aspect of this connection is that the uncertainty involved in any problem-solving situation is a result of some information deficiency, which may be incomplete, imprecise, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way (Klir and Yuan, 1995). The general framework of fuzzy reasoning allows handling much of this uncertainty, fuzzy systems employ type-1 fuzzy sets, which represents uncertainty by numbers in the range [0, 1]. When something is uncertain, like a measurement, it is difficult to determine its exact value, and of course, type-1 fuzzy sets makes more sense than using crisp sets (Zadeh, 1975). However, it is not reasonable to use an accurate membership function for something uncertain, so in this case what we need is another type of fuzzy sets, those which are able to handle these uncertainties, the so called type-2 fuzzy sets (Mendel, 2000). So, the amount of uncertainty in a system can be reduced by using type-2 fuzzy logic because it offers better capabilities to handle linguistic uncertainties by modeling vagueness and

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 133–144, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

unreliability of information (Karnik and Mendel, 2001). In industry, type-2 fuzzy logic and neural networks was used in the control of non-linear dynamic plants (Hagras, 2004) (Melin and Castillo, 2004).

This chapter deals with the optimization of interval type-2 membership functions in a fuzzy logic controller (FLC). Since, uncertainty is inherent in controllers for real world applications, as a first step, we are presenting how to deal with it using type-2 FLC to diminish the effects of imprecise information. We are supporting this statement with experimental results, qualitative observations, and quantitative measures of errors. For quantifying the errors, we utilized three widely used performance criteria, these are: Integral of Square Error (ISE), Integral of the Absolute value of the Error (IAE), and Integral of the Time multiplied by the Absolute value of the Error (ITAE) (Deshpande and Ash, 1988). Then as a second step, we optimized the parameters of the Gaussian membership functions (MFs) using the Human Evolutionary Model (HEM) which will be explained in section 11.3, and ISE as the fitness function. In this case, we used as an output, the average of two type-1 system.

11.2 Fuzzy Controllers

A FLS, described completely in terms of type-1 fuzzy sets is called a type-1 fuzzy logic system (type-1 FLS). It is composed by a knowledge base that comprises the information given by the process operator in form of linguistic control rules; a fuzzification interface, who has the effect of transforming crisp data into fuzzy sets; an inference system, that uses them in conjunction with the knowledge base to make inference by means of a reasoning method; and a defuzzification interface, which translates the fuzzy control action so obtained to a real control action using a defuzzification method.

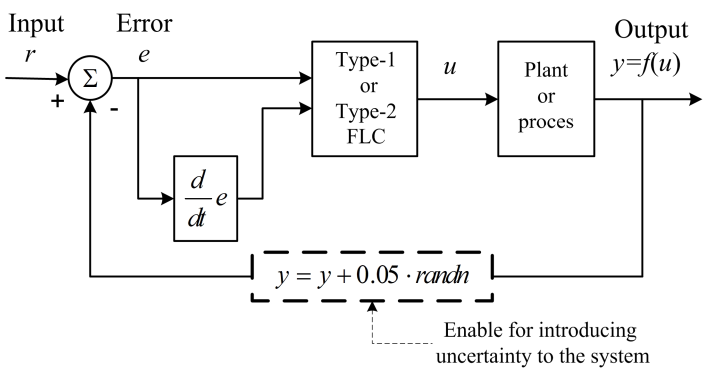


Fig. 11.1. System used for obtaining the experimental results

In this chapter, the implementation of the fuzzy controller in terms of type-1 fuzzy sets, has two input variables such as the error $e(t)$, the difference between the reference signal and the output of the process, as well as the error variation $\Delta e(t)$,

$$e(t) = r(t) - y(t) \tag{11.1}$$

$$\Delta e(t) = e(t) - e(t-1) \tag{11.2}$$

so the control system can be represented as in Fig. 11.1.

A FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain. On the other hand, type-2 FLSs, are very useful in circumstances where it is difficult to determine an exact certainty, and measurement uncertainties (Mendel, 2000).

Similar to a type-1 FLS, a type-2 FLS includes fuzzifier, rule base, fuzzy inference engine, and output processor. The output processor includes type-reducer and defuzzifier; it generates a type-1 fuzzy set output (from the type-reducer) or a crisp number (from the defuzzifier) (Mendel, 2000). A type-2 FLS is again characterized by IF-THEN rules, but its antecedent or consequent sets are now type-2. In the case of the implementation of the type-2 FLC, we have the same characteristics as in type-1 FLC, but we used type-2 fuzzy sets as membership functions for the inputs and for the output.

For evaluating the transient closed-loop response of a computer control system we can use the same criteria that normally are used for adjusting constants in PID (Proportional Integral Derivative) controllers. These are:

1. Integral of Square Error (ISE).

$$\text{ISE} = \int_0^{\infty} [e(t)]^2 \, dt \tag{11.3}$$

2. Integral of the Absolute value of the Error (IAE).

$$\text{IAE} = \int_0^{\infty} | e(t) | \, dt \tag{11.4}$$

3. Integral of the Time multiplied by the Absolute value of the Error (ITAE).

$$\text{ITAE} = \int_0^{\infty} t \, | e(t) | \, dt \tag{11.5}$$

The selection of the criteria depends on the type of response desired, the errors will contribute different for each criterion, so we have that large errors will increase the value of ISE more heavily than to IAE. ISE will favor responses with smaller overshoot for load changes, but ISE will give longer settling time. In ITAE, time appears as a factor, and therefore, ITAE will penalize heavily errors that occur late in time, but virtually ignores errors that occurs early in time.

11.3 The Human Evolutionary Model

The main idea of this computational model (Montiel et al., 2007),is to combine synergetically diferent techniques for performing search and optimization tasks. HEM was defined as follows (Montiel et al., 2007):

$$HEM = (H, AIIS, P, O, S, E, L, TL / PS, VRL, POS)$$

where

H	Human
AIIS	Adaptive Intelligent Intuitive System
P	Population of size *N* individuals
O	Single or a multiple objective optimization goals
S	Evolutionary strategy used for reaching the objectives expressed in *O*
E	Environment, here we can have predators, etc.
L	Landscape, i.e., the scenario where the evolution must be performed
TL/S	Tabu List formed by the bests solutions found/Pareto Set
VRL	Visited Regions List
POS	Pareto Optimal Set

Fig. 11.2 is a schematic representation of one individual which is comprised of three parts: a genetic representation *gr*, which can be codified using binary or floating-point representation; a set of genetic effects *ge*, that are attributes of each individual such as “physical structure”, “gender”, “actual age”, “maximum age allowed”, pheromone level”, etc; these attributes give to the algorithm some of the human like characteristics that will define in great part, the individual behavior.

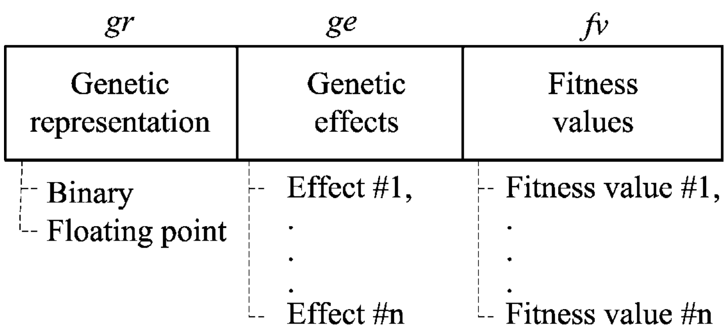


Fig. 11.2. Representing one individual in *HEM*

The third part in the individual representation is devoted to individual’s fitness values. An individual p_i is defined as $p_i=(gr_i,ge_i,fv_i)$ where $gr_i=(gr_{i1},\ldots,gr_{iM})$ is a vector (a row) of the matrix *GR* of dimension $M \times N$. The genetic effects (ge_i) are rows in a matrix *GE*. In this method we can have one or several fitness values (fv), so we can handle single objective optimization problems (SOOP), and multi-objective optimization problems (MOOP). Fitness values are defined as vectors fv_i in the matrix $FV_{J\times N}$, in this way we have $fv=(fv_1,\ldots,fv_J)$. In this context, a population P_i is defined as $P_i=(GR_i+GE_i+FV_i)$. In the attribute $ge_{igender}$, we have the valid values set $\{M,F,0\}$, in this set *M* alludes a subpopulation of male individuals, *F* is used for the female subpopulation, and *0* means that this attribute will not be considered. The genetic at

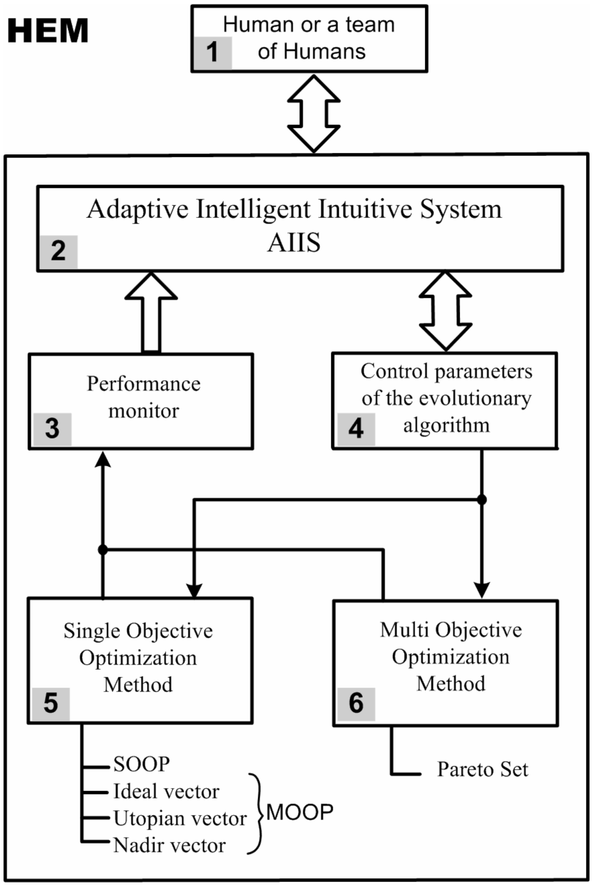


Fig. 11.3. General structure of HEM

tribute $ge_{iactAge}$ contains the actual age of an individual; its value corresponds to the number of generation that the individual has survived. We can set the maximum life expectance for each individual in the attribute $ge_{imaxAge}$. The task of the attribute $ge_{iphLevel}$ is to leave trace about which individuals have been involved in previous generations producing good offsprings. Fig. 11.3 shows the general structure of *HEM*.

In Figure 11.3, we have a general description of HEM containing six main blocks. In the first block, we show that the human or group of humans is part of the system. HEM is an intelligent evolutionary algorithm that learns from experts their rational and intuitive procedures that they use to solve optimization problems. In this model, we consider that we have two kinds of humans: real human beings and artificial humans. In the first block of Figure 11.3 we show that real human beings form one class. In the second block, the artificial human implemented in the AIIS of HEM is shown. Humans as part of the system are in charge of teaching the artificial human all the knowledge needed for realizing the searching task. The AIIS should learn the rational and intuitive knowledge from the experts; the final purpose is that the artificial human eventually can substitute the human beings most of the times. HEM has a feedback control system formed by blocks three and four; they work coordinately for monitoring and evaluating the evolution of the problem to be solved. In the fifth block, we have a single objective optimization (SOO) method for solving single objective optimization problems (SOOP). In addition, using the SOO method we can to find the ideal, utopian and nadir vectors for multiple objective optimization problems (MOOP). In the sixth block, we have a multiple objective optimization (MOO) method, which is dedicated to find the Pareto optimal set (POS) in MOOP.

11.4 Experimental Results

Figure 11.1 shows, the feedback control system that was used for achieving the results of this paper. It was implemented in Matlab where the controller was designed to follow the input as closely as possible.

The plant was modeled using equation (11.6)

$$y(i) = 0.2 \cdot y(i-3) \cdot 0.07 y(i-2) + 0.9 \cdot y(i-1) + \\ 0.05 \cdot u(i-1) + 0.5 \cdot u(i-2) \tag{11.6}$$

The controller's output was applied directly to the plant's input. Since we are interested in comparing the performance between normal type-1 and type-2 FLC system versus optimized type-2 FLC system, we tested the controller in three ways:

1. Considering the system as ideal, that is, we did not introduce in the modules of the control system any source of uncertainty. See experiments 1, and 2.
2. Simulating the effects of uncertain modules (subsystems) response introducing some uncertainty, and diverse noise levels. See experiments 3, 4 and 5.
3. After optimization of the interval type-2 MFs, we repeated case two above. See experiment 6.

For case one, as is shown in Fig. 11.1, the system's output is directly connected to the summing junction, but in the second case, the uncertainty was simulated introducing random noise with normal distribution (the dashed square in Fig. 1). We added noise to the system's output $y(i)$ using the Matlab's function "randn" which generates random numbers with Gaussian distribution. The signal and the added noise in turn, were obtained with the programmer's expression (11.7), the result $y(i)$ was introduced to the summing junction of the controller system. Note that in (11.7) we are using the value 0.05, for experiments 3 and 4, but in the set of tests for experiment 5 we varied this value to obtain different SNR values.

$$y(i) = y(i) + 0.05 \cdot randn \tag{11.7}$$

We tested the system using as input, a unit step sequence free of noise, $r(i)$. For evaluating the system's response and compare between type 1 and type 2 fuzzy controllers, we used the performance criteria ISE, IAE, and ITAE. In Table 11.3, we summarized the values obtained for each criterion considering 200 units of time. For calculating ITAE we considered a sampling time $T_s = 0.1$ sec.

For experiments 1, 2, 3, and 4 the reference input r is stable and noisy free. In experiments 3 and 4, although the reference appears clean, the feedback at the summing junction is noisy since we introduced deliberately noise for simulating the overall existing uncertainty in the system, in consequence, the controller's inputs $e(t)$ (error), and $\Delta e(t)$ contains uncertainty data.

In experiment 5, we tested the systems, type-1 and type-2 FLCs, introducing diverse values of noise η, this is modifying the signal to noise ratio SNR (Ingle and Proakis, 2000),

$$SNR = \frac{\sum |s|^2}{\sum |\eta|^2} = \frac{P_{signal}}{P_{noise}} \tag{11.8}$$

Because many signals have a very wide dynamic range, SNRs are usually expressed in terms of the logarithmic decibel scale, SNR(db),

$$SNR(db) = 10 \log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{11.9}$$

In Table 11.4, we show, for different values of SNR(db), the behavior of ISE, IAE, ITAE for type-1 and type-2 FLCs. In almost all the cases the results for type-2 FLC are better than type-1 FLC.

In type-1 FLC, we selected Gaussian MFs for the inputs and for the output. A Gaussian MF is specified by two parameters {c,σ}:

$$\mu_A(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \tag{11.10}$$

c represents the MFs center and σ determines the MFs standard deviation.

For each input of the type-1 FLC, $e(t)$ and $\Delta e(t)$, we defined three type-1 fuzzy Gaussian MFs: negative, zero, positive. The universe of discourse for these membership functions is in the range [-10 10]; their centers are -10, 0 and 10 respectively, and their standard deviations are 9, 2 and 9 respectively.

For the output of the type-1 FLC, we have five type-1 fuzzy Gaussian MFs: NG, N, Z, P and PG. They are in the interval [-10 10], their centers are -10, -4.5, 0, 4, and 10 respectively; and their standard deviations are 4.5, 4, 4.5, 4 and 4.5. Table 11.1 illustrates the characteristics of the inputs and output of the FMs of the type-1 FLC.

For the type-2 FLC, as in type-1 FLC we also selected Gaussian MFs for the inputs and for the output, but in this case we have an interval type-2 Gaussian MFs with a fixed standard deviation, σ, and an uncertain center, ie.,

$$\mu_A(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \qquad c \in [c_1, c_2] \tag{11.11}$$

Table 11.1. Characteristics of the MFs of the inputs and output of the type-1 FLC

Variable	Term	Center c	Standard Deviation σ
Input e	negative	-10	9
	zero	0	2
	positive	10	9
Input Δe	negative	-10	9
	zero	0	2
	positive	10	9
Output cde	NG	-10	4.5
	N	-4.5	4
	Z	0	4.5
	P	4	4
	PG	10	4.5

In terms of the upper and lower membership functions, we have for $\overline{\mu}_{\tilde{A}}(x)$,

$$\overline{\mu}_{\tilde{A}}(x) = \begin{cases} N(c_1, \sigma; x) & x < c_1 \\ 1 & c_1 \le x \le c_2 \\ N(c_2, \sigma; x) & x > c_2 \end{cases} \tag{11.12}$$

and for the lower membership function $\underline{\mu}_{\tilde{A}}(x)$,

$$\underline{\mu}_{\tilde{A}}(x) = \begin{cases} N(c_2, \sigma; x) & x \le \dfrac{c_1 + c_2}{2} \\ N(c_1, \sigma; x) & x > \dfrac{c_1 + c_2}{2} \end{cases} \tag{11.13}$$

where $N(c_1, \sigma, x) \equiv e^{-\frac{1}{2}\left(\frac{x - c_1}{\sigma}\right)^2}$,

and $N(c_2, \sigma, x) \equiv e^{-\frac{1}{2}\left(\frac{x - c_2}{\sigma}\right)^2}$.

Hence, in type-2 FLC, for each input we defined three interval type-2 fuzzy GaussianMFs: negative, zero, positive in the interval [-10 10], as illustrates Fig. 11.4 for input *e*. For computing the output we have five interval type-2 fuzzy Gaussian MFs NG, N, Z, P and PG, with uncertain center and fixed standard deviations in the interval [-10 10], as can be seen in Fig. 11.5. Table 11.2 shows the characteristics of the MFs of the inputs and output of the type-2 FLC.

In experiment 6, to simulate the interval type-2 MFs of the FLC, we used two type-1 FLCs. Using HEM as the optimization method, ISE as a fitness function, we found the best values, see table V, for the MFs of the inputs of these controllers. Through

Table 11.2. Characteristics of the MFs of the inputs and output of the type-2 FLC

Variable	Term	Center c_1	Center c_2	Standard Deviation σ
Input e	negative	-10.25	-9.75	9.2
	zero	-0.25	0.25	2.2
	positive	9.75	10.25	9.2
Input Δe	negative	-10.25	-9.75	9.2
	zero	-0.25	0.25	2.2
	positive	9.75	10.25	9.2
Output cde	NG	-10.25	-9.75	4.5
	N	-4.75	-4.5	4
	Z	-0.25	0.25	4.5
	P	3.75	4.25	4
	PG	9.75	10.25	4.5

an average of the two type-1 optimized FLCs, we repeated experiment 5, and calculated again the values of ISE, IAE and ITAE, as can be seen in table 11.6.

For the experiments with interval type-2 MFs not optimized, we used, basically, the type-2 toolbox that we developed.

Experiment 1. Ideal system using a type-1 FLC.
In this experiment, we did not add uncertainty data to the system. The system trends to stabilize with time and the output will follow accurately the input. In Table 11.3, we listed the obtained values of ISE, IAE, and ITAE for this experiment.

Experiment 2. Ideal system using a type-2 FLC.
Here, we used the same test conditions of Experiment 1, but in this case, we implemented the controller's algorithm with type-2 fuzzy logic. The corresponding performance criteria are listed in Table 11.3.

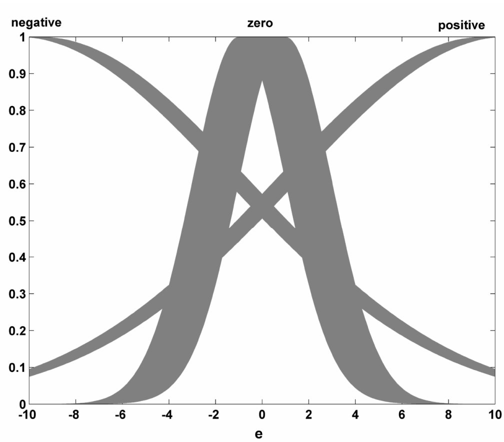


Fig. 11.4. Input *e* membership functions for the type-2 FLC

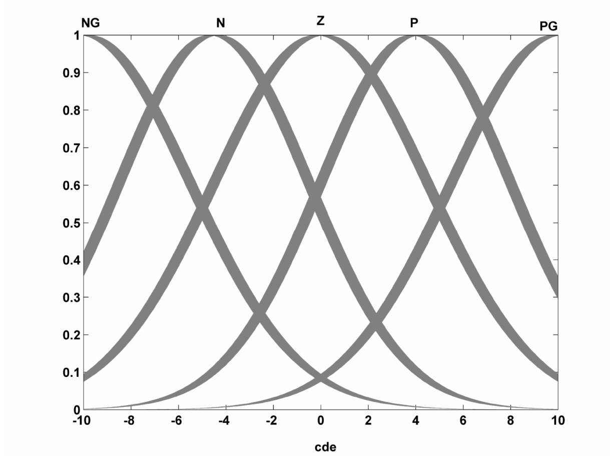


Fig. 11.5. Output *cde* membership functions fot the type-2 FLC

Experiment 3. System with uncertainty using a type-1 FLC.
In this case, we simulated using equation (7), the effects of uncertainty introduced to the system by transducers, amplifiers, and any other element that in real world applications affects expected values. In Table 11.3, we can see the obtained values of ISE, IAE, and ITAE for a simulated 10 db signal noise ratio.

Experiment 4. System with uncertainty using a type-2 FLC. In this experiment, we introduced uncertainty in the system, in the same way as in Experiment 3. In this case, we used a type-2 FLC and we improved those results obtained with a type-1 FLC in Experiment 3, see table 11.3.

Table 11.3. Comparison of performance criteria for type-1 and type-2 fuzzy logic controllers for 10 db signal noise ratio. values obtained after 200 samples.

Performance Criteria	**Type-1 FLC**		**Type-2 FLC**	
	Ideal System	Syst. with uncertainty	Ideal System	Syst. with uncertainty
ISE	5.2569	205.0191	5.2572	149.3097
IAE	13.8055	155.9412	13.7959	131.77
ITAE	46.0651	1583.4	45.8123	1262.2

Experiment 5. Varying the signal to noise ratio in type-1 and type-2 FLCs.
To test the robustness of the type-1 and type-2 FLCs, we repeated experiments 3 and 4 giving different noise levels, going from 30 db to 6 db of SNR ratio in each experiment. In Table 11.4, we summarized the values for ISE, IAE, ITAE considering 200

Table 11.4. Behavior of type -1 and type-2 fuzzy logic controllers after variation of signal noise ratio. Values obtained for 200 samples.

SNR	Type-1 FLC			Type-2 FLC		
db	ISE	IAE	ITAE	ISE	IAE	ITAE
6	1208	392.3	4903	1113	368.76	4388
8	1004	352.4	4526	903	330.38	4104
10	205.0	155.9	1583.4	149.3	131.77	1262.2
12	89.77	102.1	974.97	89.8	102.21	974.93
14	56.47	80.88	769.51	56.78	80.85	770.25
16	36.28	64.36	610.86	36.39	64.21	610.65
18	23.76	51.54	485.19	23.81	51.32	485.16
20	16.14	41.75	386.45	16.04	41.59	386.6
22	11.36	34.65	310.04	11.25	34.28	308.87
24	8.54	29.25	249.67	8.39	28.78	247.89
26	6.87	25.15	202.46	6.72	24.78	201.07
28	5.9	22.2	166.38	5.78	21.92	165.21
30	5.38	20.12	139.27	5.27	19.77	137.47

Table 11.5. Characteristics of the optimized MFs of the inputs and output of the type-2 FLC

Variable	**Term**	**Center c_1**	**Standard Deviation σ_1**	**Center c_2**	**Standard Deviation σ_2**
Input e	negative	-10	9	-10	8.0298
	zero	0	2	0	1.0987
	positive	10	9	10	8.1167
Input Δe	negative	-10	9.2	-10	8.7767
	zero	0	2.2	0	1.0987
	positive	10	9.2	10	8.5129
Output cde	NG	-10	4.5	-10	4.5
	N	-4.5	4	-4.5	4
	Z	0	4.5	0	4.5
	P	4	4	4	4
	PG	10	4.5	10	4.5

units of time with a P_{signal} of 22.98 db in all cases. As it can be seen in Table 11.4, in presence of diverse noise levels, the behavior of type-2 FLC is better than type-1 FLC above 10 db.

From Table 11.4, taking two examples, the extreme cases; we have for an SNR ratio of 8 db, in type-1 FLC the next performance values ISE=1004, IAE=352.45, ITAE=4526; for the same case, in type-2 FLC, we have ISE=903, IAE=330.38, ITAE=4104.

For 10 db of SNR ratio, we have for type-1 FLC, ISE=205.01, IAE=155.94, ITAE=1583.4, and for type-2 FLC, ISE=149.3, IAE=131.77, ITAE=1262.2.

These values indicate a better performance of type-2 FLC than type-1 FLC above certain noise values, because they are a representation of the errors and as bigger they are the performance of the system is worst

To optimize the interval type-2 MFs of the FLC, we simulated the system using two type-1 FLCs . We maintain constant the centers of the Gaussian MFs of the inputs and varied its standard deviations.

After using HEM as the optimization method, and taking ISE as the fitness function, we found the best values of the MFs, as can be seen in table 11.5. With the new values of the MFs of both type-1 FLCs, we repeated experiment 5, but in this case, we used the average of the two type-1 FLCs as the output of the type-2 system. Table 11.6, shows the results for this experiment, as can be seen, all the values of ISE were improved, and in general we can see that the performance of the system is better.

Experiment 6. Optimizing the interval type-2 MFs of the FLC.

Table 11.6. Comparison of performance criteria for type-1 FLC, and type-2 fuzzy logic controller with optimized MFs, for 10 db signal noise ratio. Values obtained after 200 samples.

SNR	**Type-1 FLC**			**Type-2 FLC**		
db	ISE	IAE	ITAE	ISE	IAE	ITAE
6	1208	392.3	4903	616.4	274.7	3005
8	1004	352.4	4526	437.3	226.7	2509
10	205.0	155.9	1583.4	115	116.6	1119.6
12	89.77	102.1	974.97	72.8	90.9	866.8
14	56.47	80.88	769.51	45.6	71.3	674.1
16	36.28	64.36	610.86	28.9	56.3	528.4
18	23.76	51.54	485.19	18.6	45.2	419.4
20	16.14	41.75	386.45	12.6	37	337
22	11.36	34.65	310.04	8.9	30.8	273.8
24	8.54	29.25	249.67	6.8	26.3	227.7
26	6.87	25.15	202.46	5.6	23.1	195.6
28	5.9	22.2	166.38	4.9	21	172.8
30	5.38	20.12	139.27	4.5	19.6	157.8

11.5 Summary

We observed and quantified using performance criteria such as ISE, IAE, and ITAE that in systems without uncertainties (ideal systems) is a better choice to select a type-1 FLC since it works a little better than a type-2 FLC, and it is easier to implement it. It is known that type-1 FLC can handle nonlinearities, and uncertainties up to some extent.

Unfortunately, real systems are inherently noisy and nonlinear, since any element in the system contributes with deviations of the expected measures because of thermal noise, electromagnetic interference, etc., moreover, they add nonlinearities from element to element in the system.

In the simulation of real systems, systems with uncertainty, we observed that the results presented in Table 11.4 demonstrated that the performance of this kind of controllers is better under high noise levels. After optimizing the interval type- 2 MFs the performance of the system is improved as we can see in table 11.6.

We can say that using a type-2 FLC in real world applications can be a better choice since this type of system is a more suitable system to manage uncertainty, as we can see in the results shown in tables 11.5 and 11.6.

12 Design of Fuzzy Inference Systems with the Interval Type-2 Fuzzy Logic Toolbox

This chapter presents the development and design of a graphical user interface and a command line programming Toolbox for construction, edition and simulation of Interval Type-2 Fuzzy Inference Systems. The Interval Type-2 Fuzzy Logic System Toolbox (IT2FLS), is an environment for interval type-2 fuzzy logic inference system development. Tools that cover the different phases of the fuzzy system design process, from the initial description phase, to the final implementation phase, constitute the Toolbox. The Toolbox's best qualities are the capacity to develop complex systems and the flexibility that allows the user to extend the availability of functions for working with the use of type-2 fuzzy operators, linguistic variables, interval type-2 membership functions, defuzzification methods and the evaluation of Interval Type-2 Fuzzy Inference Systems.

12.1 Introduction

On the past decade, fuzzy systems have displaced conventional technologies in different scientific and system engineering applications, especially in pattern recognition and control systems. The same fuzzy technology, in approximation reasoning form, is resurging also in the information technology, where it is now giving support to decision-making and expert systems with powerful reasoning capacity and a limited quantity of rules. The fuzzy sets were presented by L.A. Zadeh in 1965 to process / manipulate data and information affected by un-probabilistic uncertainty/imprecision (Zadeh, 1975). These were designed to mathematically represent the vagueness and uncertainty of linguistic problems; thereby obtaining formal tools to work with intrinsic imprecision in different type of problems; it is considered a generalization of the classic set theory. Intelligent Systems based on fuzzy logic are fundamental tools for nonlinear complex system modeling. Fuzzy sets and fuzzy logic are the base for fuzzy systems, where their objective has been to model how the brain manipulates inexact information. Type-2 fuzzy sets are used for modeling uncertainty and imprecision in a better way. These type-2 fuzzy sets were originally presented by Zadeh in 1975 and are essentially "fuzzy fuzzy" sets where the fuzzy degree of membership is a type-1

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 145–154, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

fuzzy set (Karnik and Mendel, 1998). The new concepts were introduced by Mendel and Liang allowing the characterization of a type-2 fuzzy set with a inferior membership function and an superior membership function; these two functions can be represented each one by a type-1 fuzzy set membership function (Mendel, 2000). The interval between these two functions represents the footprint of uncertainty (FOU), which is used to characterize a type-2 fuzzy set. The uncertainty is the imperfection of knowledge about the natural process or natural state. The statistical uncertainty is the randomness or error that comes from different sources as we use it in a statistical methodology. Type-2 fuzzy sets have been applied to a wide variety of problems by (Melin and Castillo, 2004).

12.2 Interval Type-2 Fuzzy Set Theory

A type-2 fuzzy set (Liang and Mendel, 2000) expresses the non-deterministic truth degree with imprecision and uncertainty for an element that belongs to a set. A type-2 fuzzy set denoted by $\tilde{\tilde{A}}$, is characterized by a type-2 membership function $\mu_{\tilde{\tilde{A}}}(x,u)$, where x∈X, u ∈ $J_x^u \subseteq [0,1]$ and $0 \leq \mu_{\tilde{\tilde{A}}}(x,u) \leq 1$ is defined in equation (1).

$$\tilde{\tilde{A}} = \left\{(x, \mu_{\tilde{\tilde{A}}}(x)) \mid x \in X\right\} = \left\{ \int_{x \in X} \left[\int_{u \in J_x^u \subseteq [0,1]} f_x(u)/u \right] / x \right\} \tag{12.1}$$

An example of a type-2 membership function constructed in the IT2FLS Toolbox was composed by a Pi primary and a Gbell secondary type-1 membership functions, these are depicted in Figure 12.1.

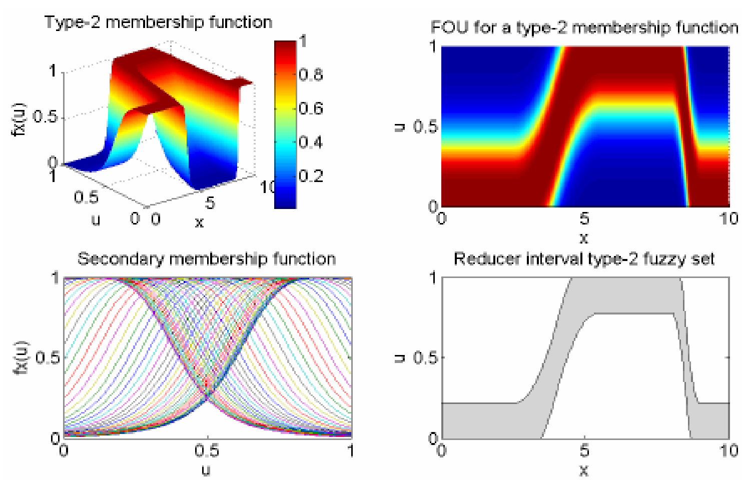


Fig. 12.1. FOU for Type-2 Membership Functions

If $f_x(u) = 1, \forall u \in [\underline{J}_x^u, \overline{J}_x^u] \subseteq [0,1]$, the type-2 membership function $\mu_{\tilde{A}}(x,u)$ is expressed by one inferior type-1 membership function, $\underline{J}_x^u \equiv \underline{\mu}_A(x)$ and one superior type-1 membership function, $\overline{J}_x^u \equiv \overline{\mu}_A(x)$ (Fig. 2), then it is called an interval type-2 fuzzy set (Mendel, 2001) denoted by equations (12.2) and (12.3).

$$\tilde{\tilde{A}} = \left\{ \int_{x \in X} \left[\int_{u \in [\underline{\mu}_A(x), \overline{\mu}_A(x)] \subseteq [0,1]} 1/u \right] / x \right\} \tag{12.2}$$

If $\tilde{\tilde{A}}$ is a type-2 fuzzy Singleton, the membership function is defined by equation (12.3)

$$\mu_{\tilde{A}}(x) = \begin{cases} 1/1 & si\ x = x' \\ 1/0 & si\ x \neq x' \end{cases} \tag{12.3}$$

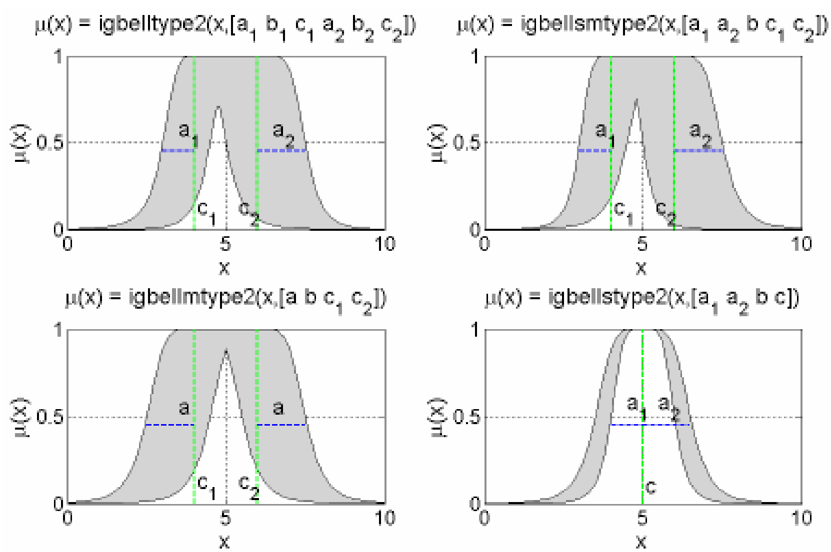


Fig. 12.2. FOU for Gbell Primary Interval Type-2 Membership Functions

Interval Type-2 Fuzzy Inference System

The human knowledge is expressed as a set of fuzzy rule. The fuzzy rules are basically of the form IF <Antecedent> THEN <Consequent> and expresses a fuzzy relationship

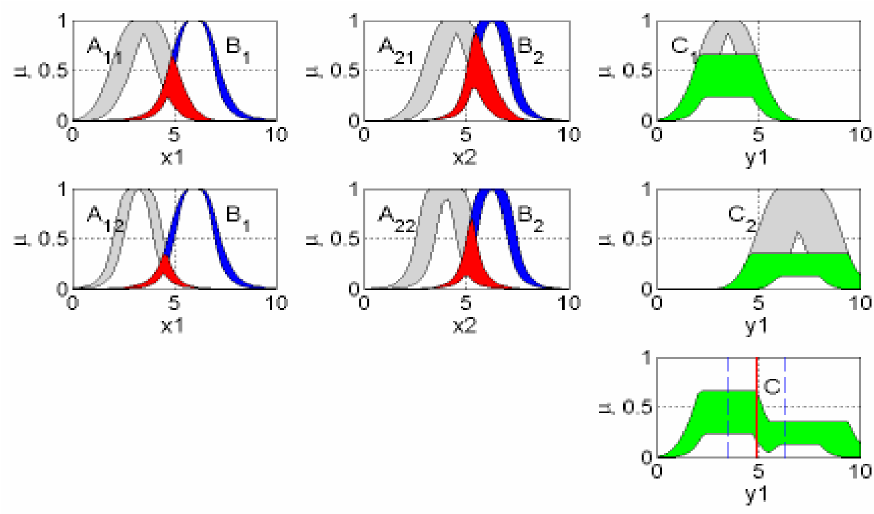


Fig. 12.3. Interval Type-2 Fuzzy Reasoning

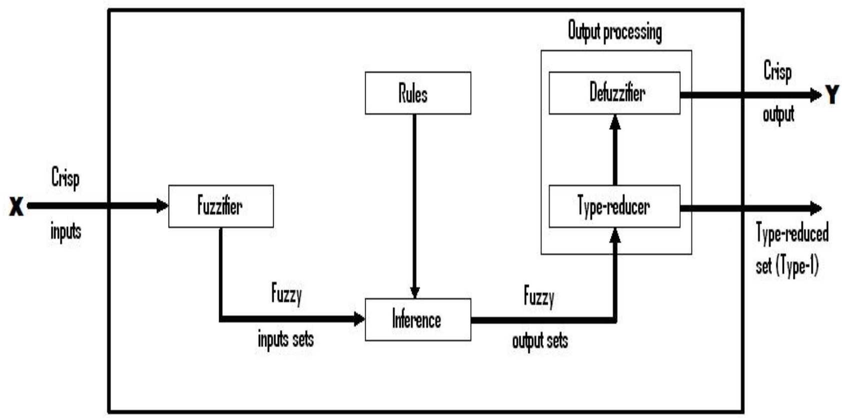


Fig. 12.4. Type-2 inference fuzzy system structure

or proposition. In fuzzy logic the reasoning is imprecise, it is approximated, which means that we can infer from one rule a conclusion even if the antecedent doesn't comply completely. We can count on two basic inference methods between rules and inference laws, Generalized Modus Ponens (GMP) (Yager, 1983) (Zadeh, 1989) and Generalized Modus Tollens (GMT) that represent the extensions or generalizations of classic reasoning. The GMP inference method is known as direct reasoning and is resumed as:

Rule	*IF x is A THEN y is B*
Fact	*x is A'*
Conclusion	*y is B'*

Where A, A', B and B' are fuzzy sets of any kind. This relationship is expressed as $B' = A' \circ (A \rightarrow B)$. Figure 13.3 shows an example of Interval Type-2 direct reasoning with Interval Type-2 Fuzzy Inputs. An Inference Fuzzy System is a rule base system that uses fuzzy logic, instead of Boolean logic utilized in data analysis (Yager, 1980) (Zadeh, 1988). Its basic structure includes four components (Fig. 13.4).

12.3 Interval Type-2 Fuzzy System Design

The Mamdani and Takagi-Sugeno-Kang Interval Type-2 Fuzzy Inference Models (Mendel, 2001) and the design of Interval Type-2 membership functions and operators are implemented in the IT2FLS (Interval Type-2 Fuzzy Logic Systems) Toolbox which was build on top of the Matlab® commercial Fuzzy Logic Toolbox. The IT2FLS Toolbox contain the functions to create Mamdani and TSK Interval Type-2 Fuzzy Inference Systems (newfistype2.m), functions to add input-output variables and their ranges (addvartype2.m), it has functions to add 22 types of Interval Type-2 Membership functions for input-outputs (addmftype2.m), functions to add the rule matrix (addruletype2.m), it can evaluate the Interval Type-2 Fuzzy Inference Systems (evalifistype2.m), evaluate Interval Type-2 Membership functions (evalimftype2.m), it can generate the initial parameters of the Interval Type-2 Membership functions (igenparamtype2.m), it can plot the Interval Type-2 Membership functions with the input-output variables (plotimftype2.m), it can generate the solution surface of the

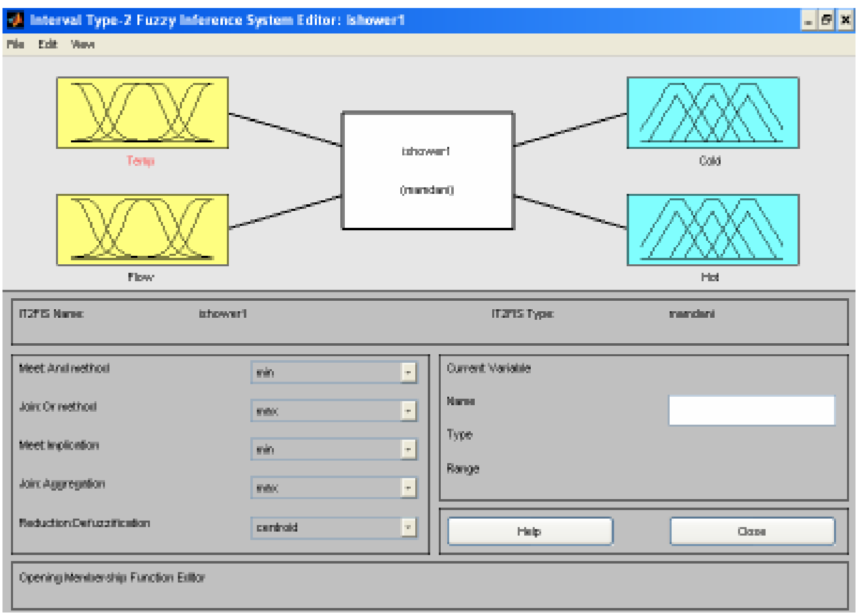


Fig. 12.5. IT2FIS Editor

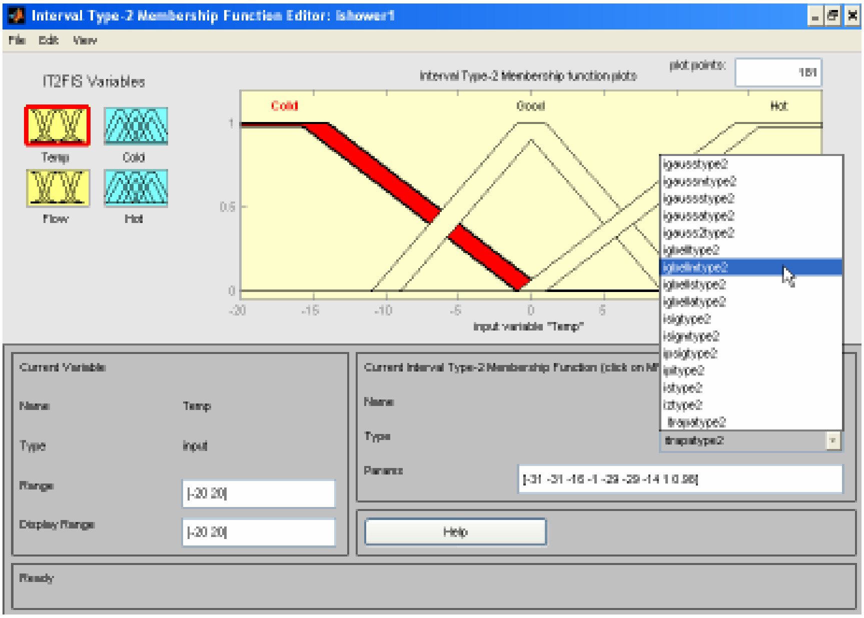


Fig. 12.6. Interval Type-2 MF's Editor

Fuzzy Inference System (gensurftype2.m), it plots the Interval type-2 membership functions (plot2dtype2.m, plot2dctype2.m), a folder to evaluate the derivatives of the Interval type-2 Membership Functions (dit2mf) and a folder with different and generalized Type-2 Fuzzy operators (it2op, t2op).

The implementation of the IT2FLS GUI is analogous to the GUI used for Type-1 FLS in the Matlab® Fuzzy Logic Toolbox, thus allowing the experienced user to adapt easily to the use of IT2FLS GUI (Castro, 2006). Figures 12.5 and 12.6 show the main viewport of the Interval Type-2 Fuzzy Inference Systems Structure Editor called IT2FIS (Interval Type-2 Fuzzy Inference Systems).

12.4 Simulation Results

We present results of a comparative analysis of the Mackey-Glass chaotic time-series forecasting study using intelligent cooperative architecture hybrid methods, with neural networks, (Mamdani, Takagi-Sugeno-Kang) type-1 fuzzy inference systems and genetic algorithms (neuro-genetic, fuzzy-genetic and neuro-fuzzy) and an interval type-2 fuzzy logic model, for the implicit knowledge acquisition in a time series behavioral data history (Castro, 2005). Also we present a shower simulation and a truck backer-upper simulation with interval type-2 fuzzy logic systems using the IT2FLS Toolbox.

Mackey-Glass chaotic time-series

To identify the model we make an exploratory series analysis with 5 delays, $L^5x(t)$, 6 periods and 500 training data values to forecast 500 output values. The IT2FLS (Takagi-Sugeno-Kang) system works with 4 inputs, 4 interval type-2 membership functions (igbellmtype2) for each input, 4 rules (Fig. 7) and one output with 4 interval linear functions, it is evaluated with no normalized values (Figure 12.7). The root mean square error (RMSE) forecasted is 0.0235. Table 12.1 shows the RMSE differences of six forecasting methods, where CANFIS and IT2FLS-TSK evaluate the best Mackey-Glass series forecasts respectively. The advantage of using the interval type-2 fuzzy logic forecasting method is that it obtains better results, even when data contains high levels of noise, furthermore we can use this method for better uncertainty series limits forecasting.

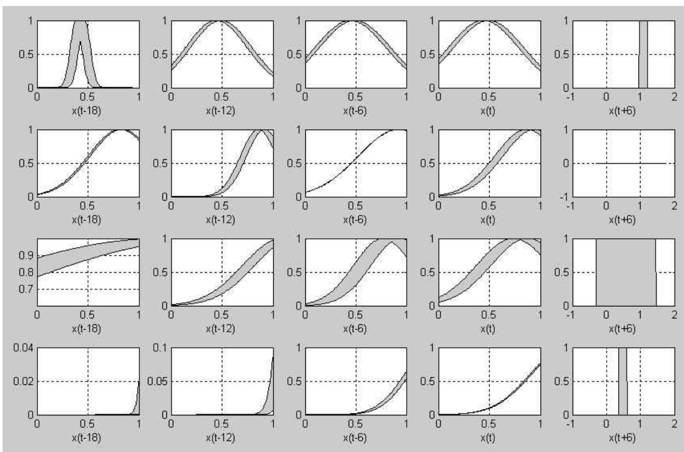


Fig. 12.7. IT2FLS (TSK) Rules

Table 12.1. Forecasting of Time Series

Methods	Mackey-Glass			
	RMSE	*trn/chk*	*epoch*	*cpu(s)**
NNFF** †	0.0595	500/500	200	13.36
CANFIS	0.0016	500/500	50	7.34
NNFF-GA†	0.0236	500/500	150	98.23
FLS(TKS)-GA†	0.0647	500/500	200	112.01
FLS(MAM)-GA†	0.0693	500/500	200	123.21
IT2FLS	0.0235	500/500	6	20.47

* POWER BOOK G4 1.5 Ghz / 512 MB RAM.

** Architecture: 4-13-1 † 30 samples average.

Shower Control Simulation

In figures 12.8 and 12.9 we compare the type-1 and type-2 fuzzy control results for the temperature and shower control simulation. The control variables signal of the interval type-2 fuzzy logic system show a better respond signal than the type-1 fuzzy logic system.

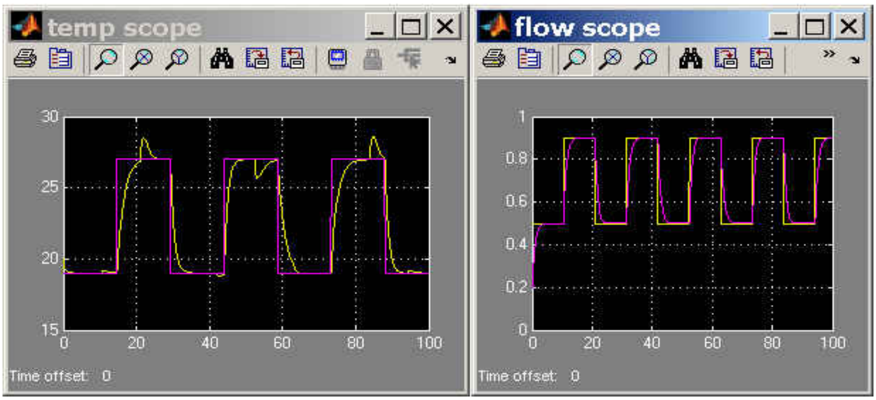


Fig. 12.8. Temperature and Flow. Type-1 fuzzy control

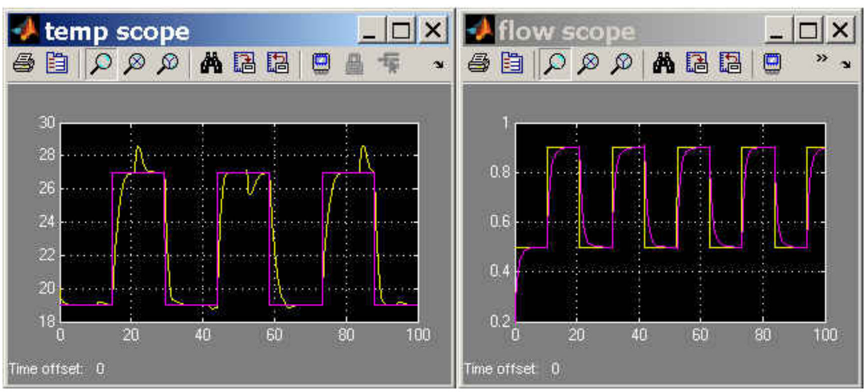


Fig. 12.9. Temperature and Flow. Interval type-2 fuzzy control

Truck backer-upper control simulation.

In figures 12.10 and 12.11 we compare the type-1 and interval type-2 fuzzy control trajectories for the truck backer-upper control simulation. In the truck backer-upper control using interval type-2 fuzzy logic, the trajectories are more stable and smooth than the type-1 fuzzy control.

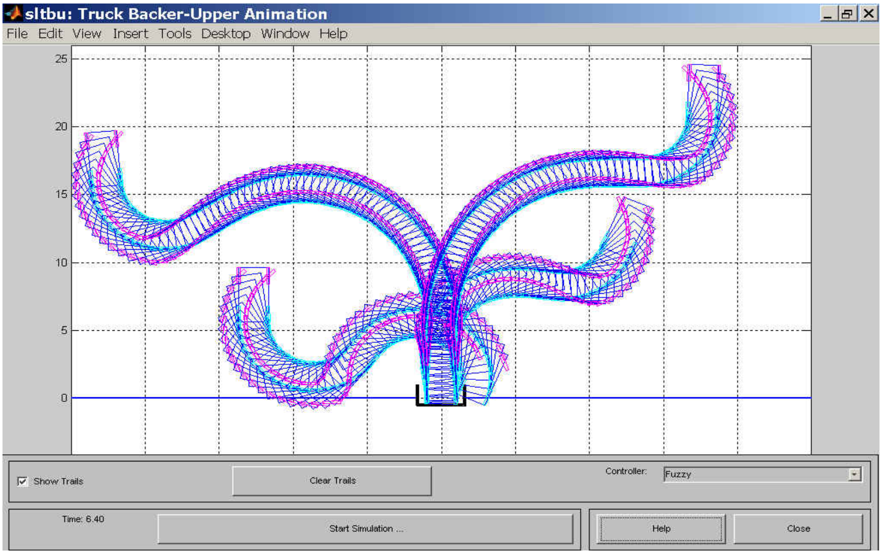


Fig. 12.10. Trajectories obtained with the type-1 fuzzy control

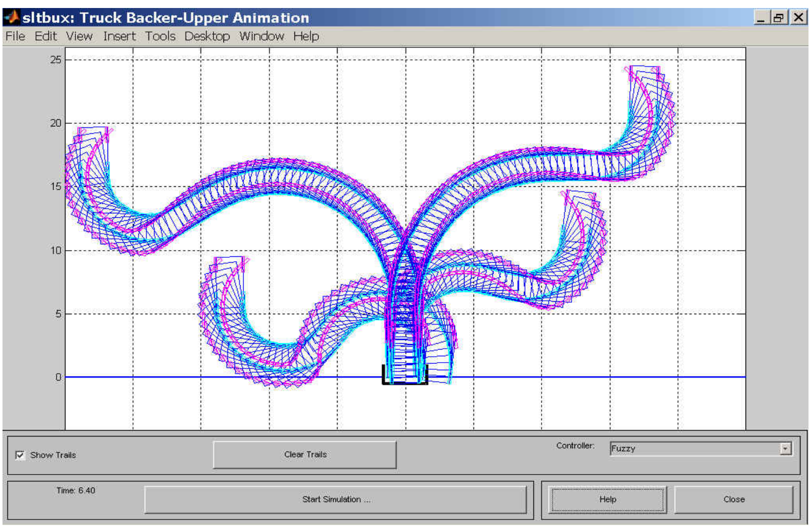


Fig. 12.11. Trajectories obtained with the interval type-2 fuzzy control

12.5 Summary

The time series results show that intelligent hybrid methods and interval type-2 fuzzy models can be derived as a generalization of the autoregressive non-lineal models in the context of time series. This derivation allows a practical specification for a general class of prognosis and identification time series model, where a set of input-output variables are part of the dynamics of the time series knowledge base. This helps the application of the methodology to a series of diverse dynamics, with a very low number of causal variables to explain behavior. The results in the interval type-2 fuzzy control cases of the shower and truck backer upper have similar results to the type-1 fuzzy control with moderate uncertain footprints. To better characterize the interval type-2 fuzzy models we need to generate more case studies with better knowledge bases for the proposed problems, therefore classify the interval type-2 fuzzy model application strengths. The design and implementation done in the IT2FLS Toolbox is potentially important for research in the interval type-2 fuzzy logic area, thus solving complex problems on the different applied areas. Our future work is to improve the IT2FLS Toolbox with a better graphics user interface (GUI) and integrate a learning technique Toolbox to optimize the knowledge base parameters of the interval type-2 fuzzy inference system and design interval type-2 fuzzy neural network hybrid models.

13 Intelligent Control of the Pendubot with Interval Type-2 Fuzzy Logic

We describe in this chapter adaptive model-based control of non-linear plants using type-2 fuzzy logic and neural networks. First, the general concept of adaptive model-based control is described. Second, the use of type-2 fuzzy logic for adaptive control is described. Third, a neuro-fuzzy approach is proposed to learn the parameters of the fuzzy system for control. A specific non-linear plant was used to simulate the hybrid approach for adaptive control. The specific plant was also used as test bed in the experiments. The non-linear plant that was considered is the "Pendubot", which is a non-linear plant similar to the two-link robot arm. The results of the type-2 fuzzy logic approach for control were good, both in accuracy and efficiency.

13.1 Introduction

Adaptive control is a method for designing a controller with some adjustable parameters and an embedded mechanism for adjusting these parameters (Castillo and Melin, 2001). Adaptive controllers have been used mainly to improve the controller's performance online (Diao and Passino, 2002). For each control cycle, the adaptive algorithm is normally implemented in three basics steps:

1. Observable data are collected to calculate the controller's performance,
2. The controller's performance is used as a guidance to calculate the adjustment to a set of controller parameters,
3. The controller's parameters are then adjusted to improve the performance of the controller.

A typical application of adaptive control has been to calibrate a system at startup. In this case, a controller is also designed for a specific class of dynamic systems (Chen and Pham, 2001). However the parameters that characterize the dynamic behavior of a particular system might not be known in advance. A controller is then designed and arbitrary values are assigned to initialize these parameters. After a few control cycles, parameters are adjusted to converge to the actual parameters of the system. This approach is often used for cases in which a system is designed to handle

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 155–170, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

a variable payload. The payload is different each time, e.g., a crane is used to pick up a sizeable object. The payload will alter the basic dynamic behavior of a dynamic system. Adaptive control is normally used to calibrate these parameters that characterize such dynamic behavior. Traditionally, there are four basic approaches for adaptive control (Castillo and Melin, 2003): (i) gain scheduling, (ii) model reference adaptive system, (iii) self-tuning regulator, and (iv) dual control. Gain scheduling is a method of adjusting the control signal based on a known look-up table describing changes of a dynamic system. The model reference adaptive system is a method of comparing the performance of the actual system against an assumed mathematical model that describes the actual system, and designing control input to drive this comparison error to zero (Melin and Castillo, 2001). Self-tuning regulator is a method of updating the parameters of a model that describes the plant based on observed data, and channeling the updated information into the controller that is designed based on these parameters. Dual control is a method of extending adaptive control to stochastic model dealing with uncertainties.

In this chapter, an extension to the principle of adaptive control using type-2 fuzzy logic is proposed. First, the concept of adaptive fuzzy control is described. A neuro-fuzzy approach is used to learn the parameters of the fuzzy system using real data from the plant. The proposed adaptive control was tested with a specific non-linear plant, to evaluate the performance of this approach. The Pendubot was used as the non-linear plant in this study (Fantoni and Lozano, 2002). The simulation and experimental results clearly show the advantage of using type-2 fuzzy logic to improve the concept of adaptive control.

13.2 Adaptive Fuzzy Control

Adaptive fuzzy control is an extension of fuzzy control theory to allow the fuzzy controller, extending its applicability, either to a wider class of uncertain systems or to fine-tune the parameters of a system to accuracy (Chen and Pham, 2001). In this scheme, a fuzzy controller is designed based on knowledge of a dynamic system (Margaliot and Langholz, 2000). This fuzzy controller is characterized by a set of parameters (Passino and Yurkovich, 1998). These parameters are either the controller constants or functions of a model's constants.

A controller is designed based on an assumed mathematical model representing a real system. It must be understood that the mathematical model does not completely match the real system to be controlled. Rather, the mathematical model is seen as an approximation of the real system. A controller designed based on this model is assumed to work effectively with the real system if the error between the actual system and its mathematical representation is relatively insignificant. However, there exists a threshold constant that sets a boundary for the effectiveness of a controller. An error above this threshold will render the controller ineffective toward the real system. An adaptive controller is set up to take advantage of additional data collected at run time for better effectiveness. At run time, data are collected periodically at the beginning of each constant time interval, $t_n = t_{n-1} + □t$, where □t is a constant measurement of time, and $[t_n, t_{n-1})$ is a duration between data collection. Let D_n be a set of data collected at time $t = t_n$. It is assumed that at any particular time, $t = t_n$, a history of data $\{D_0, D_1,$

..., D_n} is always available. The more data available, more accurate the approximation of the system will become.

At run time, the control input is fed into both the real system and the mathematical model representing the system. The output of the real system and the output of that mathematical model are collected and an error representing the difference between these two outputs are calculated. Let x(t) be the output of the real system, and y(t) the output of the mathematical model. The error ε(t) is defined as:

$$\varepsilon(t) = x(t) - y(t). \quad (13.1)$$

Figure 13.1 depicts this tracking of the difference between the mathematical model and the real dynamic system it represents.

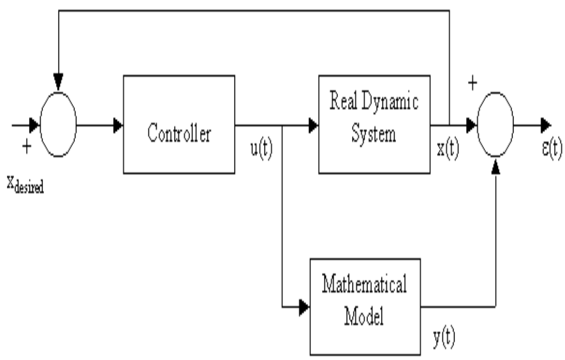


Fig. 13.1. Tracking the error function between outputs of a real system and mathematical model

An adaptive controller will be adjusted based on the error function ε(t). This calculated data will be fed into either the mathematical model or the controller for adjustment. Since the error function ε(t) is available only at run time, an adjusting mechanism must be designed to accept this error as it becomes available, i.e., it must evolve with the accumulation of data in time. At any time, $t = t_n$, the set of calculated data in the form of a time series $\{\varepsilon(t_0), \varepsilon(t_1),..., \varepsilon(t_n)\}$is available and must be used by the adjusting mechanism to update appropriate parameters.

In normal practice, instead of doing re-calculation based on a lengthy set of data, the adjusting algorithm is reformulated to be based on two entities: (i) sufficient information, and (ii) newly collected data. The sufficient information is a numerical variable representing the set of data $\{\varepsilon(t_0), \varepsilon(t_1),..., \varepsilon(t_{n-1})\}$ collected from the initial time t_0 to the previous collecting cycle starting at time $t = t_{n-1}$. The new datum $\varepsilon(t_n)$ is collected in the current cycle starting at time $t = t_n$.

An adaptive controller will operate as follows. The controller is initially designed as a function of a parameter set and state variables of a mathematical model. The parameters can be updated any time during operation and the controller will adjust itself to the newly updated parameters. The time frame is usually divided into a series of equally spaced intervals $\{[t_n,t_{n+1})|\ n = 0,1,2,...;\ t_{n+1} = t_n + \varepsilon\ t\}$. At the beginning of each time interval $[t_n,t_{n+1})$ observable data are collected and the error function $\varepsilon\ (t_n)$ is calculated. This error is used to calculate the adjustment in the parameters of the controller. New control input $u(t_n)$ for the time interval $[t_n,t_{n+1})$ is then calculated based on the newly calculated parameters and fed into both the real dynamic system under control and the mathematical model upon which the controller is designed. This completes one control cycle. The next control cycle will consist of the same steps repeated for the next time interval $[t_{n+1},t_{n+2})$, and so on.

A dynamic system (Rasband, 1990) can be described by a set of fuzzy if-then rules that correlate the input and the output. These rules include a set of parameters that is used to uniquely calculate the estimated output of a system given the inputs and current states of the system. A dynamic system is mathematically modeled as a set of Sugeno fuzzy if-then rules (Sugeno and Kang, 1988):

R^1: IF $x_1(t_n)$ is X_{11} AND …AND $x_i(t_n)$ is X_{1i}

THEN $y_1(t_{n+1}) = a_{11}x_1(t_n) + \ldots + a_{1i}x_i(t_n) + \quad + b_{11}u_1(t_n) + \ldots + b_{1i}u_j(t_n)$

… … … (2)

R^k: IF $x_1(t_n)$ is X_{k1} AND …AND $x_i(t_n)$ is X_{ki}

THEN $y_k(t_{n+1}) = a_{k1}x_1(t_n) + \ldots + a_{ki}x_i(t_n) + \quad + b_{k1}u_1(t_n) + \ldots + b_{ki}u_j(t_n)$

where $x_1, x_2, \ldots, x_i$ are the observable state variables of the real system, $y_1, y_2, \ldots, y_k$ are the calculated state variables of the mathematical model, and $u_1, u_2, \ldots, u_j$ the control inputs to both the real system and the mathematical model.

The above fuzzy mathematical model is characterized by a set of parameters $\{a_{ki}, b_{kj}\}$ These parameters will determine the behavior of the mathematical model. Theoretically, the model will approximate the real system. The more accurately the model approximates the real dynamic system, the better the controller designed based on this model behaves.

Similarly, a fuzzy controller, if designed based on the fuzzy mathematical model above, will have the following form of fuzzy rules:

R^1: IF $x_1(t_n)$ is X_{11} AND …AND $x_i(t_n)$ is X_{1I}

THEN $u_1(t_n) = k_{11}x_1(t_n) + k_{12}x_2(t_n) + \ldots + k_{1i}x_i(t_n)$

… … … (3)

R^j: IF $x_1(t_n)$ is X_{k1} AND …AND $x_i(t_n)$ is X_{ki}

THEN $u_j(t_n) = k_{j1}x_1(t_n) + k_{j2}x_2(t_n) + \ldots + k_{ji}x_i(t_n)$

In this case, the set of parameters $\{k_{ji}\}$ characterizes the controller. It can be understood intuitively that the parameter set $\{k_{ji}\}$ is designed as a function of the parameter set $\{a_{ki}, b_{kj}\}$, so that the state variables of the fuzzy model are driven to a target point with stability.

13.3 Type-2 Fuzzy Logic

Fuzzy Logic Systems are comprised of rules. Quite often, the knowledge that is used to build these rules is uncertain. Such uncertainty leads to rules whose antecedents or consequents are uncertain, which translates into uncertain antecedent or consequent membership functions (Karnik and Mendel, 1998). Type-1 fuzzy systems (Jang et. al, 1997), whose membership functions are type-1 fuzzy sets, are unable to directly handle such uncertainties (Melin and Castillo, 2002). We describe briefly in this paper, type-2 fuzzy systems (Mendel, 2001), in which the antecedent or consequent membership functions are type-2 fuzzy sets. Such sets are fuzzy sets whose membership grades themselves are type-1 fuzzy sets; they are very useful in circumstances where it is difficult to determine an exact membership function for a fuzzy set.

The concept of a type-2 fuzzy set, was introduced by Zadeh, 1975, as an extension of the concept of an ordinary fuzzy set (henceforth called a "type-1 fuzzy set"). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership grade for each element of this set is a fuzzy set in [0,1], unlike a type-1 set where the membership grade is a crisp number in [0,1]. Such sets can be used in situations where there is uncertainty about the membership grades themselves, e.g., an uncertainty in the shape of the membership function or in some of its parameters. Consider the transition from ordinary sets to fuzzy sets. When we cannot determine the membership of an element in a set as 0 or 1, we use fuzzy sets of type-1. Similarly, when the situation is so fuzzy that we have trouble determining the membership grade even as a crisp number in [0,1], we use fuzzy sets of type-2.

This does not mean that we need to have extremely fuzzy situations to use type-2 fuzzy sets. There are many real-world problems where we cannot determine the exact

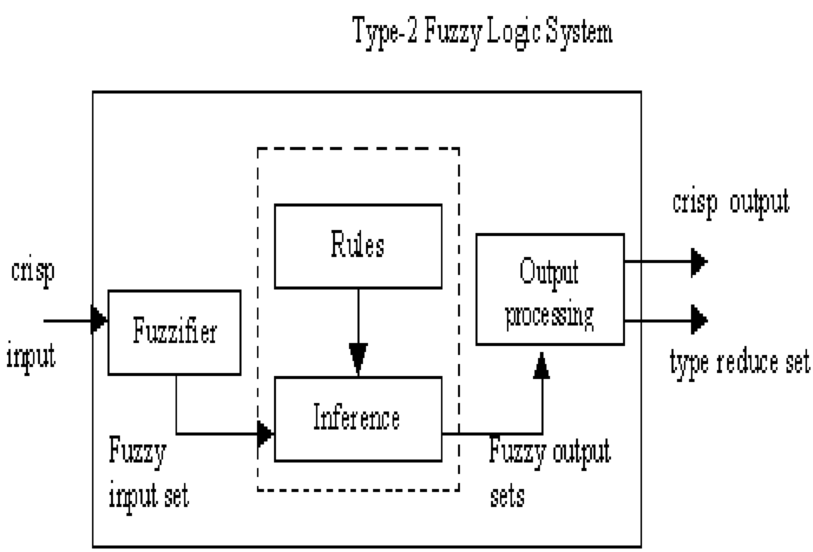


Fig. 13.2. Structure of a type-2 fuzzy system

form of the membership functions, e.g., in time series prediction because of noise in the data. Another way of viewing this is to consider type-1 fuzzy sets as a first order approximation to the uncertainty present in the real-world. Then type-2 fuzzy sets can be considered as a second order approximation. Of course, it is possible to consider fuzzy sets of higher types but the complexity of the fuzzy system increases very rapidly.

The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general, will not change for any type-n (Mendel, 2001). A higher-type number just indicates a higher "degree of fuzziness". Since a higher type changes the nature of the membership functions, the operations that depend on the membership functions change; however, the basic principles of fuzzy logic are independent of the nature of membership functions and hence, do not change. Rules of inference like Generalized Modus Ponens or Generalized Modus Tollens continue to apply.

In Figure 13.2 we show the general structure of a type-2 fuzzy system. We assume that both antecedent and consequent sets are type-2; however, this need not necessarily be the case in practice.

The structure of the type-2 fuzzy rules is the same as for the type-1 case because the distinction between type-2 and type-1 is associated with the nature of the membership functions. Hence, the only difference is that now some or all the sets involved in the rules are of type-2. In a type-1 fuzzy system, where the output sets are type-1 fuzzy sets, we perform defuzzification in order to get a number, which is in some sense a crisp (type-0) representative of the combined output sets. In the type-2 case, the output sets are type-2; so we have to use extended versions of type-1 defuzzification methods. Since type-1 defuzzification gives a crisp number at the output of the fuzzy system, the extended defuzzification operation in the type-2 case gives a type-1 fuzzy set at the output. Since this operation takes us from the type-2 output sets of the fuzzy system to a type-1 set, we can call this operation "type reduction" and call the type-1 fuzzy set so obtained a "type-reduced set". The type-reduced fuzzy set may then be defuzzified to obtain a single crisp number; however, in many applications, the type-reduced set may be more important than a single crisp number.

Type-2 sets can be used to convey the uncertainties in membership functions of type-1 fuzzy sets, due to the dependence of the membership functions on available linguistic and numerical information. Linguistic information (e.g. rules from experts), in general, does not give any information about the shapes of the membership functions. When membership functions are determined or tuned based on numerical data, the uncertainty in the numerical data, e.g., noise, translates into uncertainty in the membership functions. In all such cases, any available information about the linguistic/numerical uncertainty can be incorporated in the type-2 framework. However, even with all of the advantages that fuzzy type-2 systems have, the literature on the applications of type-2 sets is scarce. Some examples are for decision-making (Yager, 1980), and for solving fuzzy relational equations (Wagenknecht and Hartmann, 1988).

For the specific case of adaptive control of non-linear plants, the use of type-2 fuzzy rules is justified when the degree of uncertainty is high (for example, due to noise or complexity of the process) in the measured time series. Of course, a type-1 fuzzy system could be enough for adaptive control in the case of relatively simple processes in the plant (Castillo and Melin, 2001). However, in other cases there are highly non-linear processes present in the plant, like in biochemical reactors or electrochemical processes (Melin and Castillo, 2001). For this reason, we are proposing that to model the uncertainty in this type of highly non-linear processes, we need to use type-2 fuzzy logic. We illustrate the application of the type-2 fuzzy logic approach with the case of controlling the motion of the "Pendubot". Experimental results show a significant improvement in the tracking capabilities with the type-2 fuzzy logic approach.

13.4 Description of the Pendubot Plant

The Pendubot is a non-linear plant that has many attractive features for control research and education (Fantoni and Lozano, 2002). The Pendubot system consists of two links moving around a fixed point, which is very similar to the case of a two-link robot arm (Sun et. al, 2001). In Figure 3 we can appreciate a pictorial representation of the Pendubot.

The equations of motion for the Pendubot can be found using Lagrangian dynamics. In matrix form the equations are:

$$\tau = D(q)q'' + C(q,q')q' + g(q) \tag{13.4}$$

where τ is the vector of torque applied to the links and q is the vector of joint angle positions with

$$D(q) = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}$$

$$d_{11} = m_1 l^2_{c1} + m_2(l^2_1 + l^2_{c2} + 2 l_1 l_{c2} \cos q_2) + I_1 + I_2$$

$$d_{12} = d_{21} = m_2(l^2_{c2} + l_1 l_{c2} \cos q_2) + I_2$$

$$d_{22} = m_2 l^2_{c2} + I_2$$

and

$$C(q, q') = \begin{pmatrix} hq_2 & hq_2 + hq_1 \\ -hq_1 & 0 \end{pmatrix}$$

$$h = -m_2 l_1 l_{c2} \sin q_2$$

and

$$g(q) = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}$$

$$\phi_1 = (m_2 l_{c1} + m_2 l_1)\, g\cos q_1 + m_2 l_{c2}\, g\cos(q_1 + q_2)$$
$$\phi_2 = m_2 l_{c2}\, g\cos(q_1 + q_2)$$

m_1, the total mass of link one,

l_1, the length of link one,

l_{c1}, the distance to the center of mass of link one,

I_1, the moment of inertia of link one about its centroid,

m_2, the total mass of link two,

l_{c2}, the distance to the center of mass of link two,

I_2, the moment of inertia of link two about its centroid,

g, the acceleration of gravity.

From the above equations it is observed that the seven dynamic parameters can be grouped into the following five parameter equations

$$\begin{aligned}
\theta_1 &= m_1 l^2_{c1} + m_2 l^2_1 + I_1 \\
\theta_2 &= m_2 l^2_{c2} + I_2 \\
\theta_3 &= m_2 l_1 l_{c2} \\
\theta_4 &= m_1 l_{c1} + m_2 l_1 \\
\theta_5 &= m_2 l_{c2}
\end{aligned} \tag{13.5}$$

For a control design that neglects friction, these five parameters are all that are needed. There is no reason to go a step further and find the individual parameters since the control equations can be written with only the five parameters. Substituting these parameters into the above equations leaves the following matrices:

$$D(q) = \begin{pmatrix} \theta_1 + \theta_2 + 2\theta_3 \cos q_2 & \theta_2 + \theta_3 \cos q_2 \\ \theta_2 + \theta_3 \cos q_2 & \theta_2 \end{pmatrix}$$

$$C(q,q') = \begin{pmatrix} -\theta_3 \sin(q_2)q'_2 & -\theta_3 \sin(q_2)q'_2 - \theta_3 \sin(q_2)q'_1 \\ \theta_3 \sin(q_2)q'_1 & 0 \end{pmatrix}$$

$$g(q) = \begin{pmatrix} \theta_4 \, g\cos q_1 + \theta_5 \, g\cos(q_1 + q_2) \\ \theta_5 \, g\cos(q_1 + q_2) \end{pmatrix}$$

Finally, using the invertible property of the mass matrix, D(q), the state equations are given by:

$$\begin{pmatrix} q'_1 \\ q'_2 \end{pmatrix} = D(q)^{-1} \tau - D(q)^{-1} C(q,q')q - D(q)^{-1} g(q) \qquad (13.6)$$

$$x_1 = q_1, \quad x_2 = q'_1, \quad x_3 = q_2, \quad x_4 = q'_2$$

$$x_1 = x_2$$

$$x_2 = q'_1$$

$$x_3 = x_4$$

$$x_4 = q'_2$$

We show in Figure 13.4 the blocks diagram for the simulation of the Pendubot using the Simulink tool of MATLAB. Using the Simulink tool, we can simulate different dynamic behaviors of the non-linear plant and also test different intelligent controllers. We show in Figure 13.5 a sample simulation of the Pendubot plant for a specific situation.

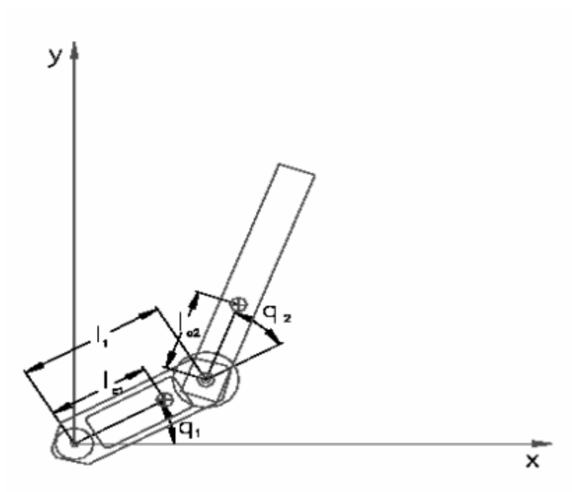


Fig. 13.3 Pictorial representation of the Pendubot non-linear plant

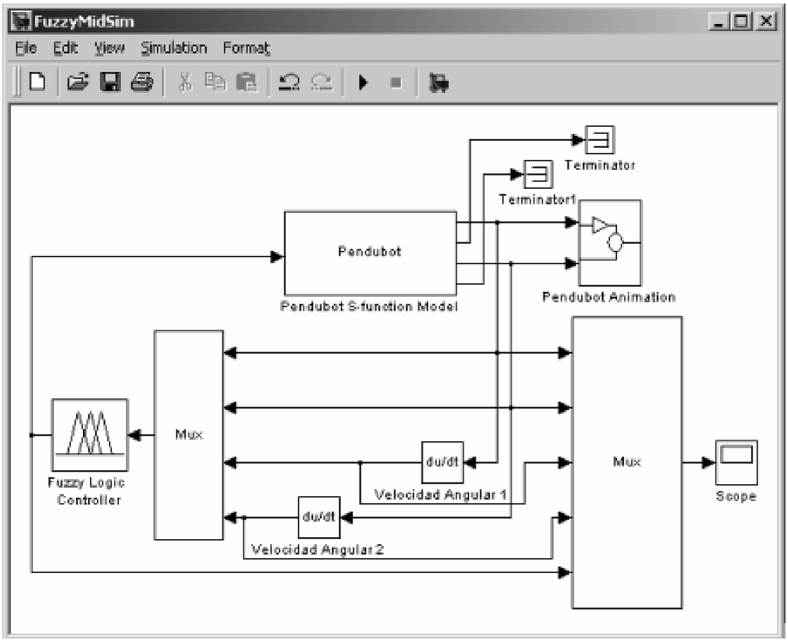


Fig. 13.4. Blocks diagram in Simulink for the Pendubot plant

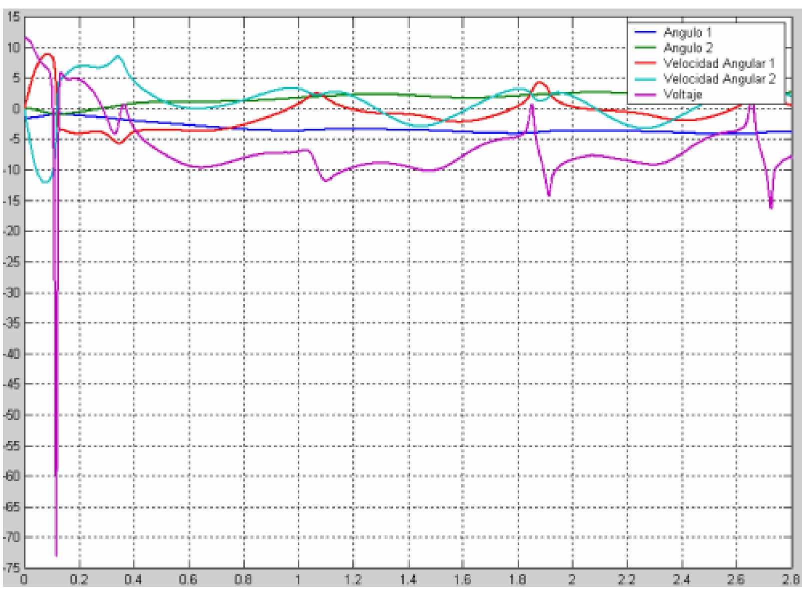


Fig. 13.5. Simulation of the Pendubot plant for a specific situation

13.5 Simulation Results with Type-1 Fuzzy Control

The Pendubot system consists of a two-link arm that can be stabilized in the upright position. The Pendubot system is one of the three configurations that can be assembled for the Mechatronics system, which is shown in real form in Figure 13.6.

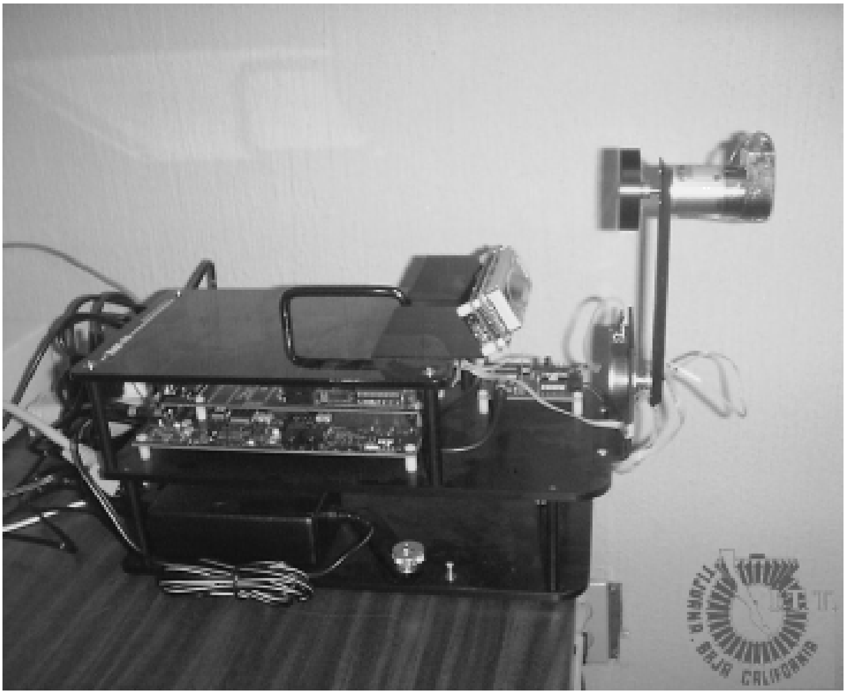


Fig. 13.6. A view of the Mechatronics system in the Control Lab

To give an idea of the performance of our neuro-fuzzy approach for adaptive model-based control of non-linear plants, we show below simulation results obtained for the Pendubot. The desired trajectory for the link was selected to be

$$q_d = t\sin(2.0t) \tag{13.7}$$

and the simulation was carried out with the initial values: $q(0) = 0.1$ $q'_1(0) = 0$.

We used three-layer neural networks (with 15 hidden neurons) with the Levenberg-Marquardt algorithm (Melin and Castillo, 2002) and hyperbolic tangent sigmoidal functions as the activation functions for the neurons (Miller et. al, 1995). We show in Figure 13.7 the function approximation achieved with the neural network for control after 9 epochs of training with a variable learning rate. The identification achieved by the neural network can be considered very good because the error has been decreased to the order of 10^{-4}. We show in Figure 13.8 the fuzzy rule base for controlling the

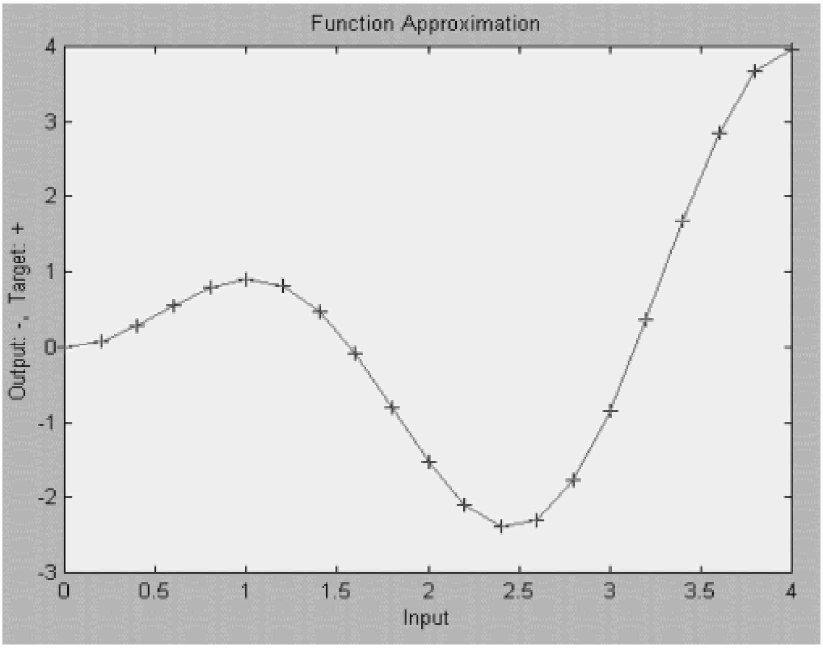


Fig. 13.7. Function approximation after 9 epochs

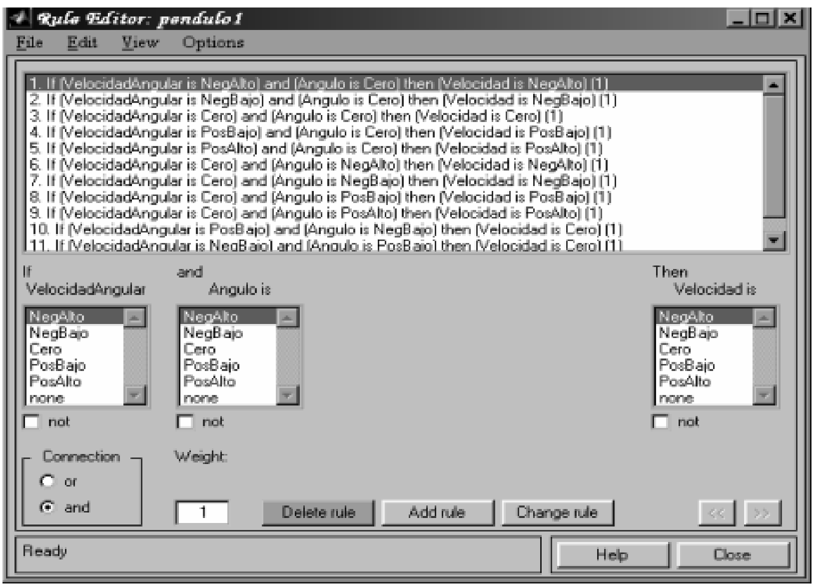


Fig. 13.8. Fuzzy rule base for control of the plant

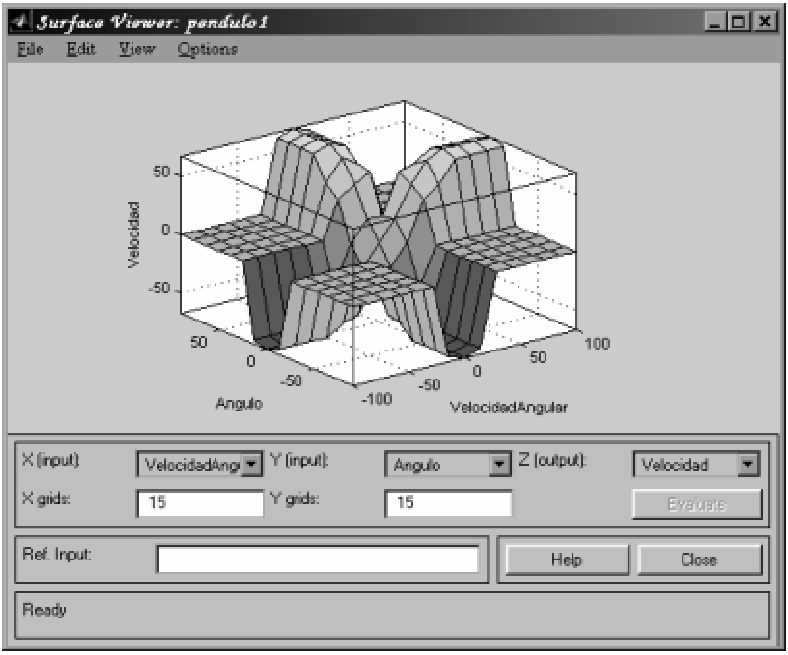


Fig. 13.9. Non-linear surface for modelling the plant

Pendubot system. The fuzzy system was implemented in the fuzzy logic toolbox of MATLAB (Nakamura, 1997). We show in Figure 13.9 the non-linear surface for controlling the non-linear plant.

Summarizing, we can say using a set of fuzzy rules for controlling the non-linear plant. However, still we can improve the performance of intelligent system for control by using a hybrid approach combining the advantages of neural networks and fuzzy logic (Zilouchian and Jamshidi, 2001). The neuro-fuzzy approach can be used to optimize the parameters of the fuzzy system (like with the ANFIS method).

13.6 Type-2 Fuzzy Control of the Pendubot

For the case of the type-2 fuzzy logic approach, we have to change our way of calculating the output of the fuzzy system. Now, we use interval computations to find the minimum and maximum values for obtaining the outputs of the type-2 fuzzy system. We basically, compute the outputs of two type-1 fuzzy systems, one for computing the minimum value and the other for the maximum value. Then, if we need to further reduce the type of the output, we can apply the traditional deffuzzification methods. Fortunately, in this way we can take advantage of the machinery that we already have for type-1 fuzzy logic, as well as the computer programs in the MATLAB programming language.

We show below, in Table 13.1, the results of a type-2 fuzzy system for controlling different conditions of the plant. We also show the corresponding values of the type-1 fuzzy logic approach. A comparison, of both approaches can be made in this way. Of course, it is easy to appreciate that in the case of type-2 fuzzy logic the output result is an interval, instead of a single numeric value. In some applications, this is closer to reality, since we are expecting something similar to a confidence interval. However, in other areas of application, like in control, a unique result is needed, so in this case we need to defuzzify again (or type-reduce the result). The centroid value shown in Table 13.1 is the result of type-reducing the interval, given by the minimum and maximum values, using the centroid method.

Table 13.1 Comparison between the type-2 and the type-1 fuzzy logic approaches

Theta	Theta1	x	x1	Type-1 F.L.	Type-2 min	Type-2 max
-0.5	-0.5	-3.0	-3.0	-21.8089	-23.4666	-21.8089
-0.4	-0.4	-2.8	-2.8	-28.2943	-33.0175	-25.3952
-0.2	-0.2	-2.5	-2.5	-14.1706	-14.6773	-13.2667
0.2	0.2	1.0	1.0	14.1706	13.1480	15.7477
0.3	0.3	1.5	1.5	21.2629	20.6327	23.9127
0.4	0.4	2.5	2.5	28.2943	26.1292	33.0175
-0.4	0.4	-1.5	2.5	-15.3708	-21.8082	-13.4062
-0.3	0.2	1.5	-2.0	-10.0538	-11.0802	-9.5352
0.45	0.3	2.35	2.45	30.2580	27.5427	32.9074
0.5	0.5	3.0	3.0	33.2449	32.7477	33.1621
-0.25	0.2	-2.75	1.0	-6.7334	-6.3145	-6.2630
-0.1	0.1	0.75	0.75	3.1987	2.4332	4.0886

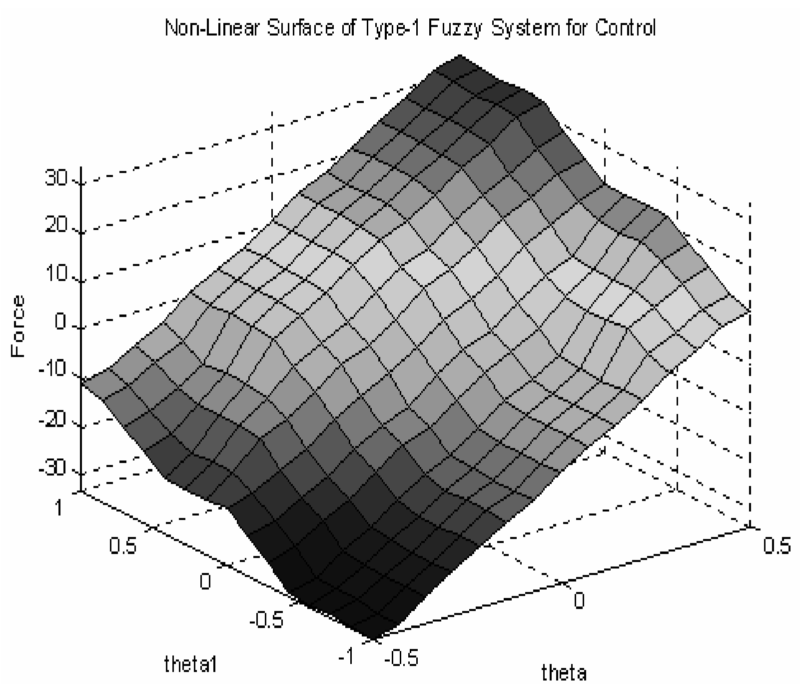


Fig. 13.10. Non-linear surface for the type-1 fuzzy system

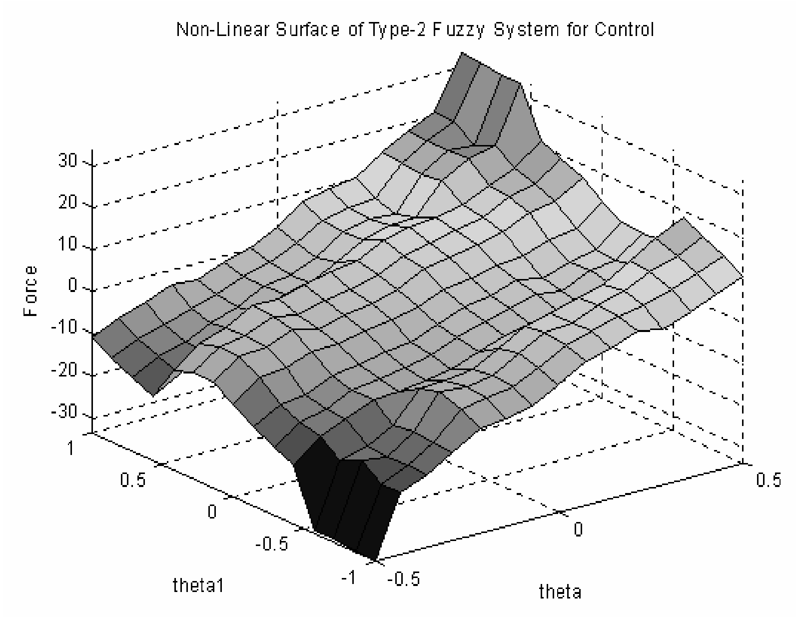


Fig. 13.11. Non-linear surface for the type-2 fuzzy system

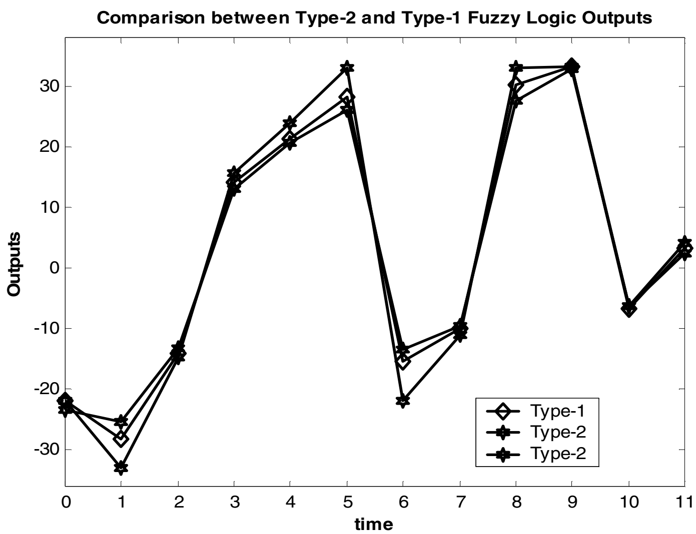


Fig. 13.12. Comparison of type-2 and type-1 fuzzy systems outputs

From Table 13.1 we can appreciate that in some cases the output of the type-2 fuzzy system is almost the same as in the type-1 case, but in other situations the results are somewhat different. For our application, we find out that the results of the type-2 fuzzy logic approach were better for controlling the plant. The main reason for saying that the type-2 approach is better, in this case, is that we have a more stable surface of control.

We show in Figures 13.10 and 13.11 the non-linear surfaces for the type-1 and type-2 fuzzy systems, respectively. From these figures, we can appreciate the difference between both fuzzy logic approaches. It is obvious that the type-2 fuzzy logic surface is smoother, which is better for controlling the plant. Finally, we show in Figure 13.12 a comparison between the predicted outputs for the type-2 and type-1 fuzzy systems. In Figure 13.12 we can appreciate that the type-1 result is almost always in between the minimum and maximum values of the type-2 fuzzy system.

13.7 Summary

We have very good simulation results for the Pendubot system for different conditions. We did compare the use of type-1 fuzzy logic for control with the use of type-2 fuzzy logic to decide which one is the best for this application. We found out that the type-2 approach was more efficient. The new method for control combines the advantages of neural networks (learning and adaptability) with the advantages of type-2 fuzzy logic (use of expert knowledge) to achieve the goal of robust adaptive control of the Pendubot system. We consider that our method for adaptive control can be applied to general non-linear dynamical systems because the hybrid approach, combining neural networks and fuzzy logic, does not depend on the particular characteristics of the non-linear dynamic plant. The main advantage of using type-2 fuzzy logic, in this case, is due to the greater ability of this theory in modeling uncertainties in the control of non-linear plants.

14 Automated Quality Control in Sound Speakers Manufacturing Using a Hybrid Neuro-fuzzy-Fractal Approach

We describe in this chapter the application of type-2 fuzzy logic to the problem of automated quality control in sound speaker manufacturing. Traditional quality control has been done by manually checking the quality of sound after production. This manual checking of the speakers is time consuming and occasionally was the cause of error in quality evaluation. For this reason, we developed an intelligent system for automated quality control in sound speaker manufacturing. The intelligent system has a type-2 fuzzy rule base containing the knowledge of human experts in quality control. The parameters of the fuzzy system are tuned by applying neural networks using, as training data, a real time series of measured sounds as given by good sound speakers. We also use the fractal dimension as a measure of the complexity of the sound signal.

14.1 Introduction

We describe in this chapter the application of a type-2 fuzzy logic approach to the problem of quality control in the manufacturing of sound speakers in a real plant. The quality control of the speakers was done before by manually checking the quality of sound achieved after production (Dickason, 1997). A human expert evaluates the quality of sound of the speakers to decide if production quality was achieved. Of course, this manual checking of the speakers is time consuming and occasionally was the cause of error in quality evaluation (Loctite, 1999). For this reason, it was necessary to consider automating the quality control of the sound speakers. The problem of measuring the quality of the sound speakers is as follows:

1. First, we need to extract the real sound signal of the speaker during the testing period after production
2. Second, we need to compare the real sound signal to the desired sound signal of the speaker, and measure the difference in some way

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 171–185, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

3. Third, we need to decide on the quality of the speaker based on the difference found in step 2. If the difference is small enough then the speaker can be considered of good quality, if not then is bad quality.

The first part of the problem was solved by using a multimedia kit that enable us to extract the sound signal as a file, which basically contains 108000 points over a period of time of 3 seconds (this is the time required for testing). We can say that the sound signal is measured as a time series of data points (Castillo and Melin, 2003), which has the basic characteristics of the speaker. The second part of the problem was solved by using a neuro-fuzzy approach to train a fuzzy model with the data from the good quality speakers (Mandelbrot, 1987). We used a neural network (Jang et al.,1997) to obtain a Sugeno fuzzy system (Sugeno and Kang, 1988) with the time series of the ideal speakers. In this case, a neural network (Rumelhart et al., 1986) is used to adapt the parameters of the fuzzy system with real data of the problem. With this fuzzy model, the time series of other speakers can be used as checking data to evaluate the total error between the real speaker and the desired one. The third part of the problem was solved by using another set of type-2 fuzzy rules (Zadeh, 1975), which basically are fuzzy expert rules to decide on the quality of the speakers based on the total checking error obtained in the previous step. Of course, in this case we needed to define type-2 membership functions for the error and quality of the product, and the Mamdani reasoning approach was used. We also use as input variable of the fuzzy system the fractal dimension of the sound signal. The fractal dimension (Mandelbrot, 1987) is a measure of the geometrical complexity of an object (in this case, the time series). We tested our fuzzy-fractal approach for automated quality control during production with real sound speakers with excellent results. Of course, to measure the efficiency of our intelligent system we compared the results of the fuzzy-fractal approach to the ones by real human experts.

14.2 Basic Concepts of Sound Speakers

In any sound system, ultimate quality depends on the speakers (Dickason, 1997). The best recording, encoded on the most advanced storage device and played by a top-of-the-line deck and amplifier, will sound awful if the system is hooked up to poor speakers. A system's speaker is the component that takes the electronic signal stored on things like CDs, tapes and DVD's and turns it back into actual sound that we can hear.

14.2.1 Sound Basics

To understand how speakers work, the first thing you need to do is understand how sound works. Inside your ear is a very thin piece of skin called the ear-drum. When your eardrum vibrates, your brain interprets the vibrations as sound. Rapid changes in air pressure are the most common thing to vibrate your eardrum.

An object produces sound when it vibrates in air (sound can also travel through liquids and solids, but air is the transmission medium when we listen to speakers). When something vibrates, it moves the air particles around it. Those air particles in turn move the air particles around them, carrying the pulse of the vibration through the air as more and more particles are pushed farther from the source of the vibration. In this way, a vibrating object sends a wave of pressure fluctuation through the atmosphere. When the fluctuation wave reaches your ear, it vibrates the eardrum back and forth. Our brain interprets this motion as sound. We hear different sounds from different vibrating objects because of variations in:

- sound wave frequency -- A higher wave frequency simply means that the air pressure fluctuates faster. We hear this as a higher pitch. When there are fewer fluctuations in a period of time, the pitch is lower.
- air pressure level -- the wave's amplitude -- determines how loud the sound is. Sound waves with greater amplitudes move our ear drums more, and we register this sensation as a higher volume.

A speaker is a device that is optimized to produce accurate fluctuations in air pressure.

14.2.2 Making Sound

In the last section we saw that sound travels in waves of air pressure fluctuation, and that we hear sounds differently depending on the frequency and amplitude of these waves. We also learned that microphones translate sound waves into electrical signals, which can be encoded onto CDs, tapes, LPs, etc. Players convert this stored information back into an electric current for use in the stereo system.

A speaker is essentially the final translation machine -- the reverse of the microphone. It takes the electrical signal and translates it back into physical vibrations to create sound waves. When everything is working as it should, the speaker produces nearly the same vibrations that the microphone originally recorded and encoded on a tape, CD, LP, etc. Traditional speakers do this with one or more drivers. A driver produces sound waves by rapidly vibrating a flexible cone, or diaphragm. Figure 14.1 shows a typical speaker driver.

The voice coil is a basic electromagnet. An electromagnet is a coil of wire, usually wrapped around a piece of magnetic metal, such as iron. Running electrical current through the wire creates a magnetic field around the coil, magnetizing the metal it is wrapped around. The field acts just like the magnetic field around a permanent magnet: It has a polar orientation -- a "north" end and a "south" end -- and it is attracted to iron objects. But unlike a permanent magnet, in an electromagnet you can alter the orientation of the poles. If you reverse the flow of the current, the north and south ends of the electromagnet switch. This is exactly what a stereo signal does -- it constantly reverses the flow of electricity. If you've ever hooked up a stereo system, then you know that there are two output wires for each speaker -- typically a black one and a red one. Figure 14.2 shows the wire that runs through the speaker system.

Essentially, the amplifier is constantly switching the electrical signal, fluctuating between a positive charge and a negative charge on the red wire. Since electrons always flow in the same direction between positively charged particles and negatively charged particles, the current going through the speaker moves one way and then reverses and flows the other way. This alternating current causes the polar orientation of the electromagnet to reverse itself many times a second.

Fig. 14.1. A typical speaker driver, with a metal basket, heavy permanent magnet and paper diaphragm

Fig. 14.2. The wire that runs through the speaker system connects to two hook-up jacks on the driver.

So how does this fluctuation make the speaker coil move back and forth? The electromagnet is positioned in a constant magnetic field created by a permanent magnet. These two magnets -- the electromagnet and the permanent magnet -- interact with each other as any two magnets do. The positive end of the electromagnet is attracted to the negative pole of the permanent magnetic field, and the negative pole of the electromagnet is repelled by the permanent magnet's negative pole.

When the electromagnet's polar orientation switches, so does the direction of repulsion and attraction. In this way, the alternating current constantly reverses the magnet forces between the voice coil and the permanent magnet. This pushes the coil back and forth rapidly, like a piston. When the electrical current flowing through the voice coil changes direction, the coil's polar orientation reverses. This changes the magnetic forces between the voice coil and the permanent magnet, moving the coil and attached diaphragm back and forth.

When the coil moves, it pushes and pulls on the speaker cone. This vibrates the air in front of the speaker, creating sound waves. The electrical audio signal can also be interpreted as a wave. The frequency and amplitude of this wave, which represent the original sound wave, dictates the rate and distance that the voice coil moves. This, in turn, determines the frequency and amplitude of the sound waves produced by the diaphragm.

14.2.3 Chunks of the Frequency Range

In the last section we saw that traditional speakers produce sound by pushing and pulling an electromagnet attached to a flexible cone. Although drivers all work on the same concept, there is actually a wide variety in driver size and power. The basic driver types are:

1. Woofers
2. Tweeters
3. Midrange

Woofers are the biggest drivers, and are designed to produce low frequency sounds. Tweeters are much smaller units, designed to produce the highest frequencies. Midrange speakers produce a range of frequencies in the middle of the sound spectrum.

14.3 Type-2 Fuzzy Logic Systems

Fuzzy Logic Systems are comprised of rules. Quite often, the knowledge that is used to build these rules is uncertain. Such uncertainty leads to rules whose antecedents or consequents are uncertain, which translates into uncertain antecedent or consequent membership functions (Karnik and Mendel, 1998). Type-1 fuzzy systems (Castillo and Melin, 2001), whose membership functions are type-1 fuzzy sets, are unable to directly handle such uncertainties. We describe briefly in this section, type-2 fuzzy systems, in which the antecedent or consequent membership functions are type-2 fuzzy sets. Such sets are fuzzy sets whose membership grades themselves are type-1 fuzzy sets; they are very useful in circumstances where it is difficult to determine an exact membership function for a fuzzy set.

14.3.1 Type-2 Fuzzy Sets

The concept of a type-2 fuzzy set, was introduced by (Zadeh, 1975) as an extension of the concept of an ordinary fuzzy set (henceforth called a "type-1 fuzzy set"). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership grade for each element of this set is a fuzzy set in [0,1], unlike a type-1 set where the membership grade is a crisp number in [0,1]. Such sets can be used in situations where there is uncertainty about the membership grades themselves, e.g., an uncertainty in the shape of the membership function or in some of its parameters. Consider the transition from ordinary sets to fuzzy sets. When we cannot determine the membership of an element in a set as 0 or 1, we use fuzzy sets of type-1. Similarly, when the situation is so fuzzy that we have trouble determining the membership grade even as a crisp number in [0,1], we use fuzzy sets of type-2.

This does not mean that we need to have extremely fuzzy situations to use type-2 fuzzy sets. There are many real-world problems where we cannot determine the exact form of the membership functions, e.g., in time series prediction because of noise in the data. Another way of viewing this is to consider type-1 fuzzy sets as a first order approximation to the uncertainty in the real-world. Then type-2 fuzzy sets can be considered as a second order approximation. Of course, it is possible to consider fuzzy sets of higher types but the complexity of the fuzzy system increases very rapidly. Let us consider some simple examples of type-2 fuzzy sets.

Example 14.1. Consider the case of a fuzzy set characterized by a Gaussian membership function with mean m and a standard deviation that can take values in $[\sigma_1,\sigma_2]$, i.e.,

$$\mu(x)=\exp \{-½[(x-m)/\sigma]^2\}; \quad \sigma \in [\sigma_1,\sigma_2] \tag{14.1}$$

Corresponding to each value of σ, we will get a different membership curve (see Figure 14.3). So, the membership grade of any particular x (except x=m) can take any of a number of possible values depending upon the value of σ, i.e., the membership grade is not a crisp number, it is a fuzzy set.

Example 14.2. Consider the case of a fuzzy set with a Gaussian membership function having a fixed standard deviation σ, but an uncertain mean, taking values in $[m_1, m_2]$, i.e.,

$$\mu(x)=\exp \{-½[(x-m)/\sigma]^2\}; \quad m \in [m_1,m_2] \tag{14.2}$$

Again, $\mu(x)$ is a fuzzy set. Figure 14.4 shows an example of such a set.

We can formally define these two kinds of type-2 sets as follows.

Definition 14.1. Gaussian type-2
A Gaussian type-2 fuzzy set is one in which the membership grade of every domain point is a Gaussian type-1 set contained in [0,1].

Definition 14.2. Interval type-2
An interval type-2 fuzzy set is one in which the membership grade of every domain point is a crisp set whose domain is some interval contained in [0,1].

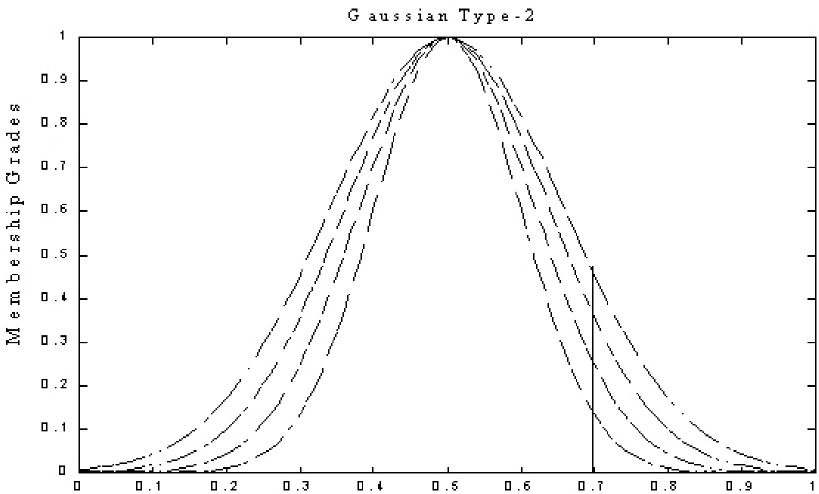


Fig. 14.3. A Type-2 fuzzy set representing a type-1 fuzzy set with uncertain standard deviation

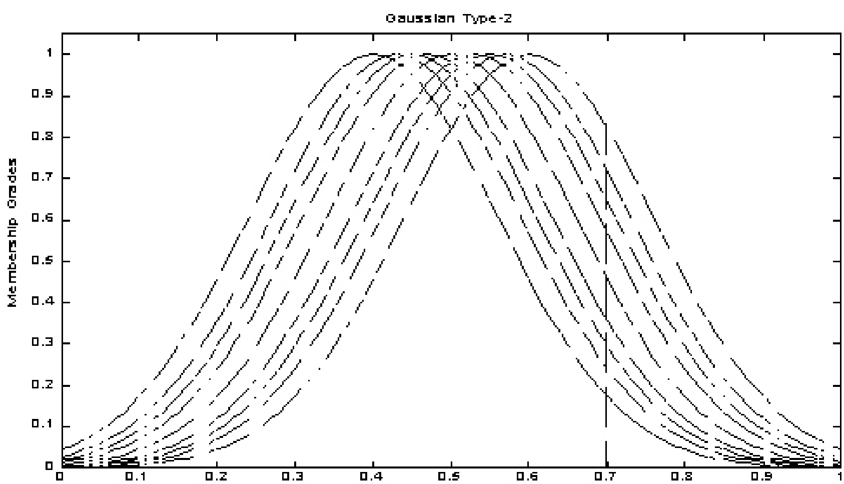


Fig. 14.4. Type-2 fuzzy set with Gaussian membership function and fixed standard deviation

14.3.2 Type-2 Fuzzy Systems

The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general, will not change for any type-n (Karnik and Mendel, 1998). A higher-type number just indicates a higher "degree of fuzziness". Since a higher type changes the nature of the membership functions, the operations that depend on the membership functions change; however, the basic principles of fuzzy logic are independent of the

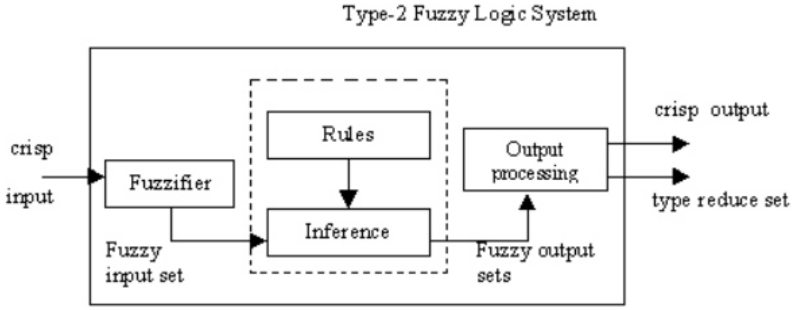


Fig. 14.5. Structure of a type-2 fuzzy system

nature of membership functions and hence, do not change. Rules of inference like Generalized Modus Ponens or Generalized Modus Tollens continue to apply. In Figure 14.5 we show the general structure of a type-2 fuzzy system. We assume that both antecedent and consequent sets are type-2; however, this need not necessarily be the case in practice.

The structure of the type-2 fuzzy rules is the same as for the type-1 case because the distinction between type-2 and type-1 is associated with the nature of the membership functions. Hence, the only difference is that now some or all the sets involved in the rules are of type-2. In a type-1 fuzzy system, where the output sets are type-1 fuzzy sets, we perform defuzzification in order to get a number, which is in some sense a crisp (type-0) representative of the combined output sets. In the type-2 case, the output sets are type-2; so we have to use extended versions of type-1 defuzzification methods. Since type-1 defuzzification gives a crisp number at the output of the fuzzy system, the extended defuzzification operation in the type-2 case gives a type-1 fuzzy set at the output. Since this operation takes us from the type-2 output sets of the fuzzy system to a type-1 set, we can call this operation “type reduction” and call the type-1 fuzzy set so obtained a “type-reduced set”. The type-reduced fuzzy set may then be defuzzified to obtain a single crisp number; however, in many applications, the type-reduced set may be more important than a single crisp number.

Type-2 sets can be used to convey the uncertainties in membership functions of type-1 fuzzy sets, due to the dependence of the membership functions on available linguistic and numerical information. Linguistic information (e.g. rules from experts), in general, does not give any information about the shapes of the membership functions. When membership functions are determined or tuned based on numerical data, the uncertainty in the numerical data, e.g., noise, translates into uncertainty in the membership functions. In all such cases, any available information about the linguistic/numerical uncertainty can be incorporated in the type-2 framework. However, even with all of the advantages that fuzzy type-2 systems have, the literature on the applications of type-2 sets is scarce. Some examples are for decision making (Yager, 1980), and for solving fuzzy relational equations (Wagenknecht and Hartmann, 1988).

14.4 Description of the Problem

The basic problem consists in the identification of sound signal quality. Of course, this requires a comparison between the real measured sound signal and the ideal good sound signal. We need to be able to accept speakers, which have a sound signal that do not differ much from the ideal signals. We show in Figure 14.6 the form of the sound signal for a good speaker (of a specific type). The measured signal contains about 108 000 points in about 3 seconds. We need to compare any other measured signal with the good one and calculate the total difference between both of them, and if the difference is small then we can accept the speaker as a good one. On the other hand, if the difference is large then we reject the speaker as a bad one.

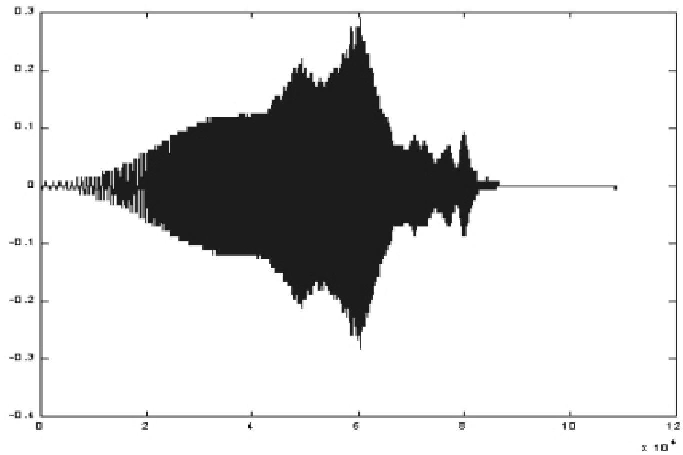


Fig. 14.6. Sound signal of a Good Speaker

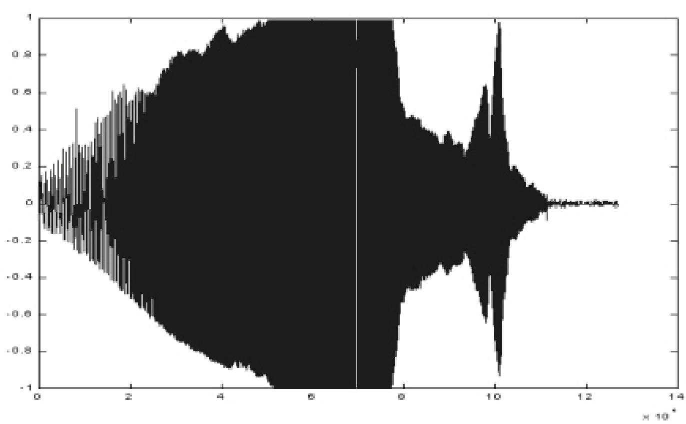


Fig. 14.7. Sound Signal of Bad Speaker (Case 1)

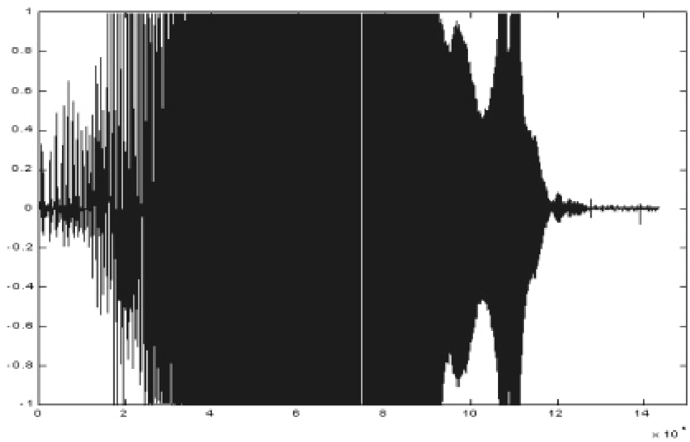


Fig. 14.8. Sound Signal of Bad Speaker (Case 2)

We show in Figure 14.7 the sound signal for a speaker of bad quality. We can clearly see the difference in the geometrical form of this signal and the one shown in Figure 14.6. In this case, the difference between the figures is sufficiently large and we easily determine that the speaker is of bad quality. We also show in Figure 14.8 another sound signal for a bad quality speaker.

14.5 Fractal Dimension of an Object

Recently, considerable progress has been made in understanding the complexity of an object through the application of fractal concepts (Mandelbrot, 1987) and dynamic scaling theory. For example, financial time series show scaled properties suggesting a fractal structure (Castillo and Melin, 2002). The fractal dimension of a geometrical object can be defined as follows:

$$d = \lim_{r \to 0} [\ln N(r)] / [\ln(1/r)] \tag{14.3}$$

where N(r) is the number of boxes covering the object and r is the size of the box. An approximation to the fractal dimension can be obtained by counting the number of boxes covering the boundary of the object for different r sizes and then performing a logarithmic regression to obtain d (box counting algorithm). In Figure 14.9, we illustrate the box counting algorithm for a hypothetical curve C. Counting the number of boxes for different sizes of r and performing a logarithmic linear regression, we can estimate the box dimension of a geometrical object with the following equation:

$$\ln N(r) = \ln\beta - d \ln r \tag{14.4}$$

this algorithm is illustrated in Figure 14.10.

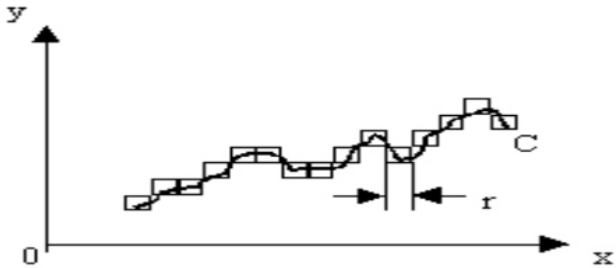


Fig. 14.9. Box counting algorithm for a curve C

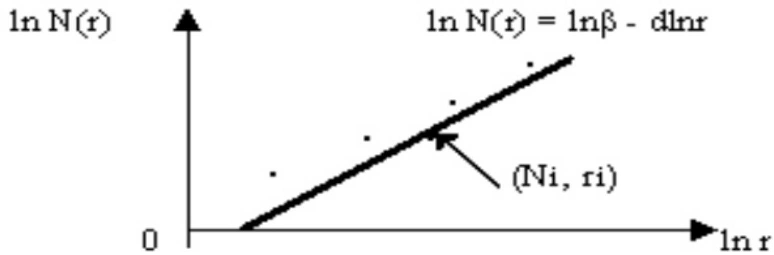


Fig. 14.10. Logarithmic regression to find dimension

We developed a computer program for calculating the fractal dimension of a sound signal. The computer program uses as input the figure of the signal and counts the number of boxes covering the object for different grid sizes. For example, the fractal dimension for the sound signal of Figure 14.6 is of 1.6479, which is a low value because it corresponds to a good speaker. On the other hand, the fractal dimension for Figure 14.7 is 1.7843, which is a high value (bad speaker). Also, for the case of Figure 14.8 the dimension is 1.8030, which is even higher (again, a bad speaker).

14.6 Experimental Results

We describe in this section the experimental results obtained with the intelligent system for automated quality control. The intelligent system uses a fuzzy rule base to determine automatically the quality of sound in speakers. We used a neural network to adapt the parameters of the fuzzy system using real data from the problem. We used the time series of 108000 points measured from a good sound speaker (in a period of 3 seconds) as training data in the neural network. We then use the measured data of any other speaker as checking data, to compare the form of the sound signals. We show in Figures 14.11 and 14.12 two cases where a neural network is used to adapt a fuzzy system with training data of good sound speakers. The approximation is very good considering the complexity of the problem. Once the training was done, we used the fuzzy system for measuring the total difference between a given signal and the good ones. This difference is used to decide on the final quality of the speaker using another set of fuzzy rules with the Mamdani approach. The type-2 fuzzy rules are as follows:

IF Difference is small AND Fractal Dimension is small THEN Quality Excellent

IF Difference is regular AND Fractal Dimension is small THEN Quality is Good

IF Difference is regular AND Fractal Dimension is high THEN Quality is Medium

IF Difference is medium AND Fractal Dimension is small THEN Quality is Medium

IF Difference is medium AND Fractal Dimension is high THEN Quality is Bad

IF Difference is large AND Fractal Dimension is small THEN Quality is Medium

IF Difference is large AND Fractal Dimension is high THEN Quality is Bad

IF Difference is small AND Fractal Dimension is high THEN Quality is Medium

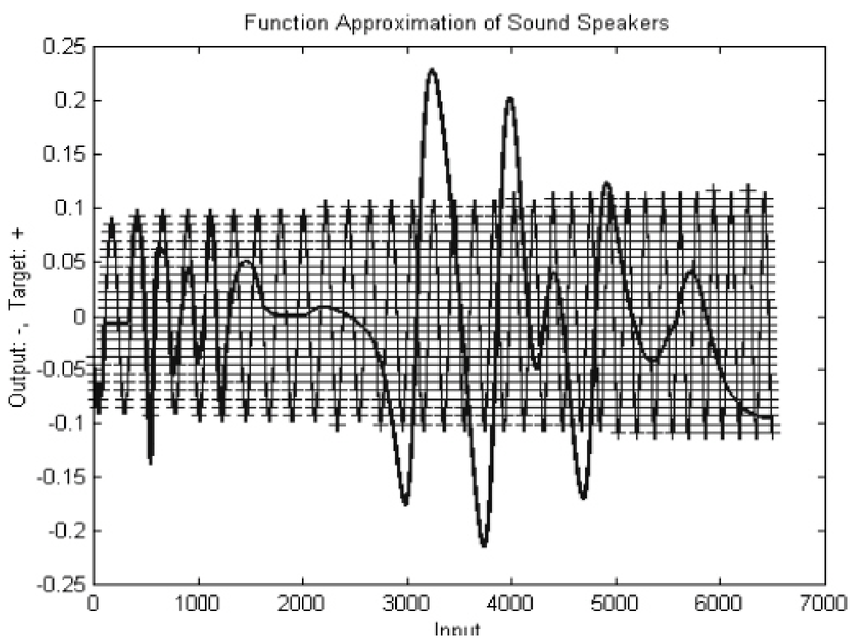


Fig. 14.11. Function approximation of the sound signal using type-2 fuzzy logic

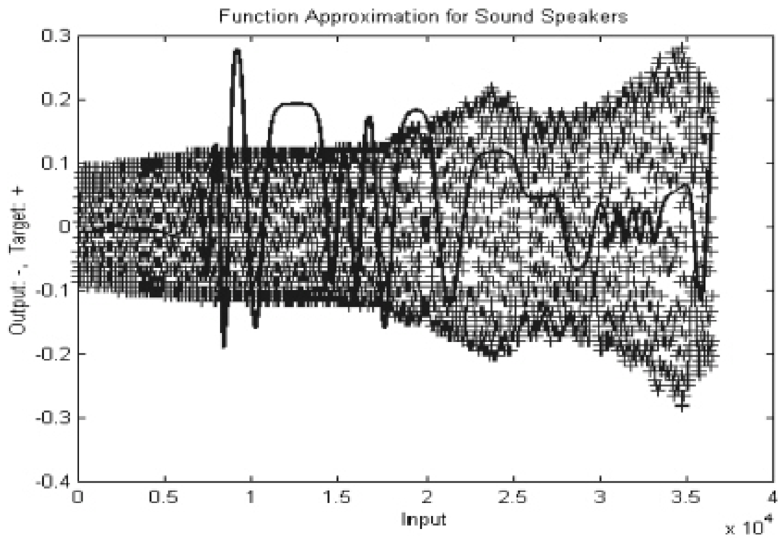


Fig. 14.12. Function approximation of the sound signal using type-2 fuzzy logic (Case 2)

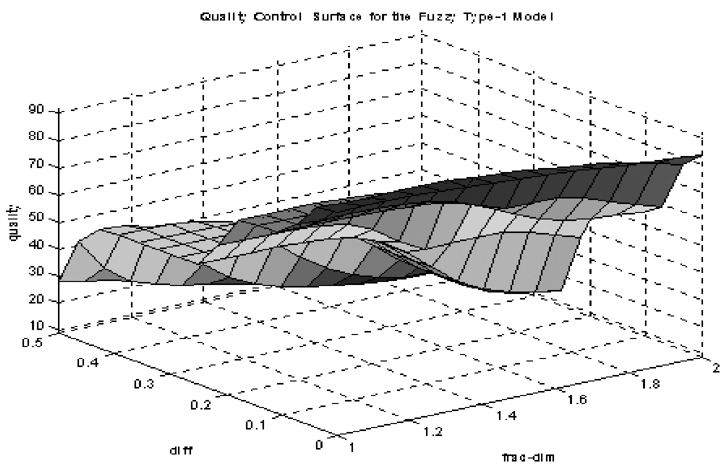


Fig. 14.13. Non-linear surface of type-1 fuzzy system

We can compare type-2 fuzzy logic with the traditional fuzzy logic in the following figures. First, we show in Figure 14.13 the non-linear surface of a type-1 fuzzy system for quality control. Second, we show in Figure 14.14 the non-linear surface of the type-2 fuzzy system for quality control. We can appreciate from these figures that the

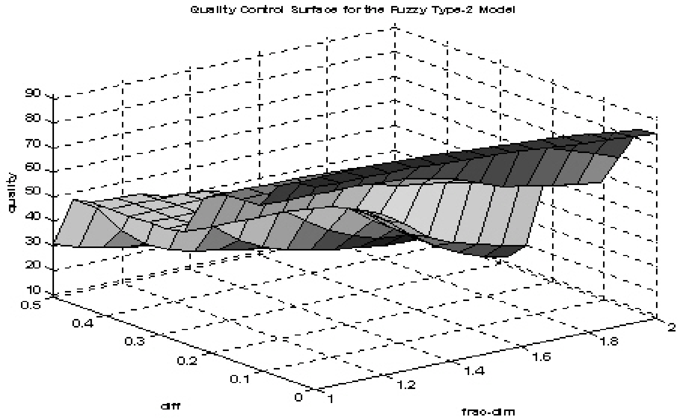


Fig. 14.14. Non-linear surface of type-2 fuzzy system

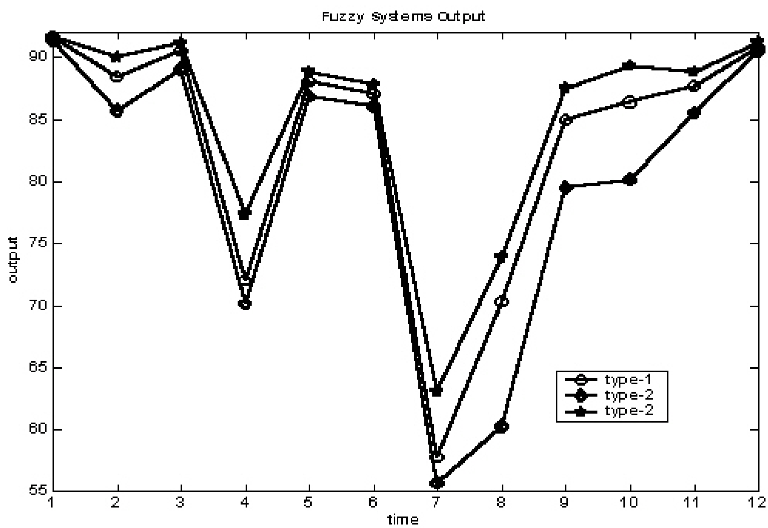


Fig. 14.15. Plot of the type-2 and type-1 fuzzy system

no-linear surface of the type-2 fuzzy system is smoother, which is due to the fact that in this case we are averaging the results of at least two type-1 fuzzy systems. We also show in Figure 14.15 (as a comparison) a plot of the results of the type-2 fuzzy system and the results of two type-1 systems for the same input data. Finally, we show in Table 14.1 the results of the type-2 and type-1 fuzzy systems for 12 specific cases.

We also show in Figure 14.16 the general architecture of the type-2 fuzzy system for quality control.

Table 14.1. Outputs of the type-2 and type-1 fuzzy systems for 12 specific situations

Input Variable		Output of the Fuzzy Systems			
Fractal Dim.	Difference	Type-1	Type-2 min	Type-2 max	Average
1.1	0.01	91.6159	91.3824	91.6843	91.05334
1.2	0.05	88.4461	85.6593	90.0590	87.8591
1.3	0.03	90.5619	89.0866	91.1582	90.1224
1.5	0.08	72.0445	70.1125	77.3158	73.7142
1.6	0.02	88.0722	86.8907	88.8715	87.8811
1.7	0.01	87.0820	86.0586	87.8339	86.9462
1.8	0.10	57.7984	55.7255	63.0281	59.3768
1.3	0.20	70.2707	60.3185	73.8411	67.0798
1.7	0.40	84.9467	79.5448	87.5151	83.5299
1.5	0.05	86.4039	80.1912	89.2951	84.7432
1.6	0.03	87.6774	85.5487	88.8180	87.1833
1.3	0.01	90.8434	90.5823	91.1990	90.8907

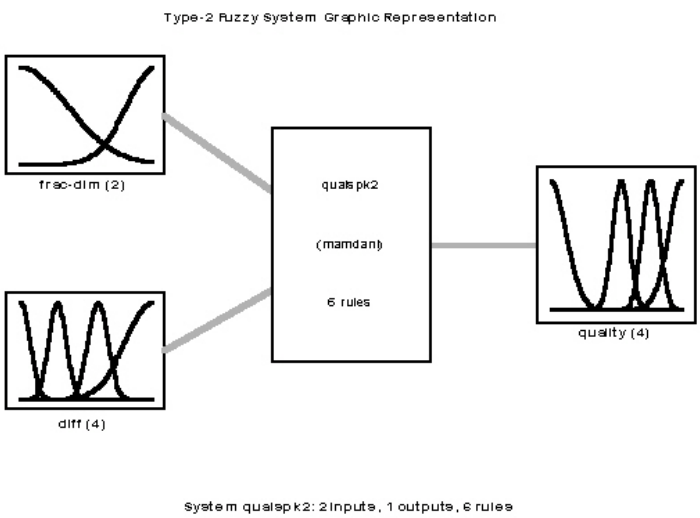


Fig. 14.16. Architecture of the fuzzy system

14.7 Summary

We described in this chapter the application of a type-2 fuzzy logic to the problem of automating the quality control of sound speakers during manufacturing in a real plant. We have implemented an intelligent system for quality control in the MATLAB programming language using the new approach. We also use the fractal dimension as a measure of geometrical complexity of the sound signals. The intelligent system performs rather well considering the complexity of the problem. The intelligent system has been tested in a real manufacturing plant with very good results.

15 A New Approach for Plant Monitoring Using Type-2 Fuzzy Logic and Fractal Theory

We describe in this chapter a new approach for plant monitoring and diagnostics using type-2 fuzzy logic and fractal theory. The concept of the fractal dimension is used to measure the complexity of the time series of relevant variables for the process. A set of type-2 fuzzy rules is used to represent the knowledge for monitoring the process. In the type-2 fuzzy rules, the fractal dimension is used as a linguistic variable to help in recognizing specific patterns in the measured data. The fuzzy-fractal approach has been applied before in problems of financial time series prediction and for other types of problems, but now it is proposed to the monitoring of plants using type-2 fuzzy logic. We also compare the results of the type-2 fuzzy logic approach with the results of using only a traditional type-1 fuzzy logic approach. Experimental results show a significant improvement in the monitoring ability with the type-2 fuzzy logic approach.

15.1 Introduction

Diagnostic systems are used to monitor the behavior of a process and identify certain pre-defined patterns that are associated with well-known problems (Du, 1998). These problems, once identified, imply suggestions for specific solutions. Most diagnostic systems are in the form of a rule-based expert system: a set of rules is used to describe certain patterns (Chiang et. al, 2000). Observed data are collected and used to evaluate these rules. If the rules are logically satisfied, the pattern is identified, and the problem associated with that pattern is suggested (Jain, et. al, 2000). In general, the diagnostic systems are used for consultation rather than replacement of human expert (Russell et. al, 2000).

Most current plant monitoring systems only check a few variables against individual upper and lower limits, and start an audible alarm should each variable move out of its predefined range (Chen and Pham, 2001). Other more complicated systems normally involve more sensors that provide more data but still follow the same pattern of independently checking individual sets of data against some upper and lower limits

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 187–202, 2008.
springerlink.com
Springer-Verlag Berlin Heidelberg 2008

(Saez and Cipriano, 2001). The warning alarm from these systems only carries a meaning that there is something wrong with the process in the plant (Yang et. al, 1999). Monitoring means checking or regulating the performance of a machine, a process, or a system (Scharf, 1991). Diagnosis, on the other hand, means deciding the nature and the cause of a diseased condition of a machine, a process, or a system by examining the symptoms. In other words, monitoring is detecting suspect symptoms, whereas diagnosis is determining the cause of the symptoms (Castillo and Melin, 2002).

In this chapter a new fuzzy-fractal approach for plant monitoring is proposed. The concept of the fractal dimension is used to measure the complexity of the time series of relevant variables for the process (Castillo and Melin, 1994). A set of type-2 fuzzy rules is used to represent the knowledge for monitoring the process (Castillo and Melin, 2001). In the type-2 fuzzy rules, the fractal dimension is used as a linguistic variable to help in recognizing specific patterns in the measured data. The fuzzy-fractal approach has been applied before in problems of financial time series prediction (Castillo and Melin, 1996) and for other types of problems (Castillo and Melin, 1998), but now it is proposed to the monitoring of plants using type-2 fuzzy logic. Fuzzy systems are comprised of rules (Yen and Langari, 1999). Quite often, the knowledge that is used to build these rules is uncertain. Such uncertainty leads to rules whose antecedents or consequents are uncertain, which translates into uncertain antecedent or consequent membership functions (Mendel, 2001). Type-1 fuzzy systems (Jang et. al, 1997), whose membership functions are type-1 fuzzy sets, are unable to directly handle such uncertainties (Wang, 1997). We describe briefly in this paper, type-2 fuzzy systems, in which the antecedent or consequent membership functions are type-2 fuzzy sets.

15.2 Monitoring and Diagnosis

Monitoring means checking or regulating the performance of a machine, a process, or a system (Du et. al, 1993). Diagnosis, on the other hand, means deciding the nature and the cause of a diseased condition of a machine, a process, or a system by examining the symptoms (Patton et. al, 2000). In other words, monitoring is detecting suspect symptoms, whereas diagnosis is determining the cause of the symptoms.

The importance of monitoring and diagnosis of plant processes now is widely recognized because it results in increased productivity, improved product quality and decreased production cost (Melin and Castillo, 2001). As a result, in the past decade, a large number of research and development projects have been carried and many monitoring and diagnosis methods have been developed (Du, 1998). The commonly used monitoring and diagnosis methods include modeling-based methods, pattern recognition methods (Yager and Filev, 1994), fuzzy systems methods (Margaliot and Langholz, 2000), knowledge-based systems methods (Melin and Castillo, 2002), artificial neural networks (Omidvar and Elliot, 1997), and genetic algorithms (Mitchell, 1998). It is interesting to note that even though these methods are rather different, they share a very similar structure as shown in Figure 15.1.

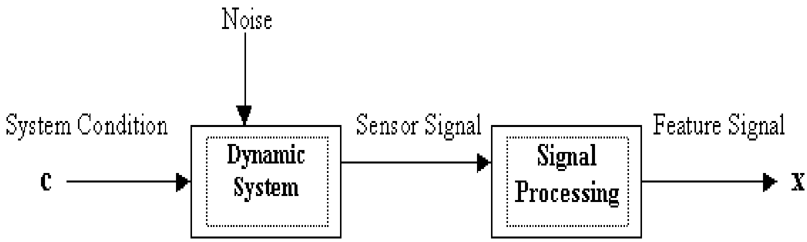


Fig. 15.1. Plant Monitoring and Diagnosis

The "health" of a machine, a process, or an engineering system (which will be referred to as system condition and denoted by $c \in \{c_1, c_2,..., c_m\}$) can be considered as the "input", the system working conditions and noises (including system noise and sampling noise) can be considered as the "noise", and the sensor signals are the "outputs" from the system. Typically, the sensor signals are processed by a computer, after which the signals are transformed into a set of features called feature signals, denoted as $\mathbf{x} = \{x_1, x_2,..., x_n\}$. In general, the systems conditions are predefined, such as normal, critical, etc. On the other hand, the features may be the mean of a temperature signal, the variance of a displacement signal, etc (Melin and Castillo, 1998). Sensing and signal processing are very important to the success of plant monitoring and diagnosis (Hsu, 2000).

More formally, the goal of monitoring is to use the feature signals, x, to determine whether the plant is in an acceptable condition(s) (a subset of $\{c_1, c_2,..., c_m\}$). On the other hand, the objective of diagnosis is to use the feature signals, x, to determine the system condition, $c \in \{c_1, c_{2,...,} c_m\}$. No matter how monitoring and diagnosis methods may differ, monitoring and diagnosis always consist of two phases: training and decision making. Training is to establish a relationship between the feature signals and the systems conditions. Without losing generality, this relationship can be represented as

$$\mathbf{x} = F(c). \tag{15.1}$$

It should be pointed out that F(c) represents a fuzzy system, a neural network or another method that could be used to obtain this relationship. In fact, it is the form of the relationship that determines the methods of monitoring and diagnosis, as well as the performance of the methods. The relationship F(c) is established based on training samples, denoted by $\mathbf{x}_1, \mathbf{x}_2,..., \mathbf{x}_k,..., \mathbf{x}_N$, where the system condition for each training sample is known [and denoted as $c(\mathbf{x}_k)$]. The conditions on F(c), so that monitoring or decision making is successful, are that F(c) is one-to-one and bounded. In this way, we can always obtain the inverse relationship, which is needed below for achieving decision making.

After the relationship is established, when a new sample is given (from an unknown system condition), its corresponding condition is estimated based on the inverse relationship

$$c = F^{-1}(\mathbf{x}). \tag{15.2}$$

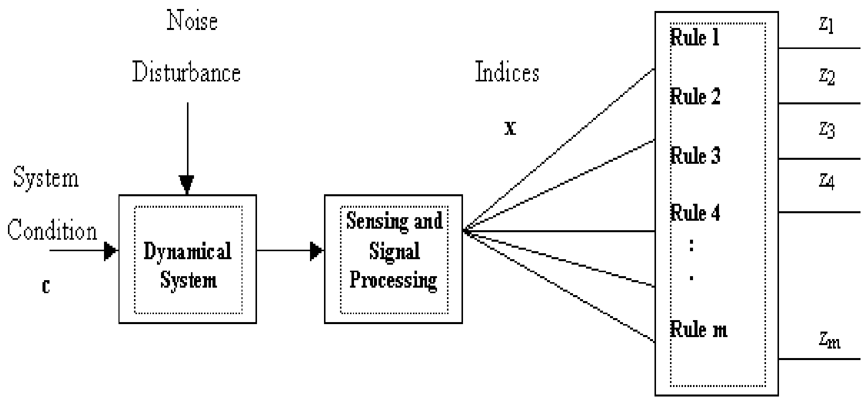


Fig. 15.2. Fuzzy system for plant monitoring and diagnosis

This is called decision-making, or classifying. Whereas it is not likely that the training samples will cover all possible cases, decision making often involves reason ing or inferencing. In particular, when a fuzzy system is used, the relationship is given by a set of fuzzy rules as shown in Figure 15.2. The input to the fuzzy system is the feature signal and the output of the fuzzy system is the estimated plant condition(s) [i.e., $\mathbf{z} = (z_1, z_2, ..., z_m)$ is an estimate of $\mathbf{c} = (c_1, c_2, ..., c_m)$]. In other words, the fuzzy system models the inverse relationship between the system conditions and the feature signals.

15.3 Fractal Dimension of a Geometrical Object

Recently, considerable progress has been made in understanding the complexity of an object through the application of fractal concepts (Mandelbrot, 1987) and dynamic scaling theory (Mandelbrot, 1997). For example, financial time series show scaled properties suggesting a fractal structure (Castillo and Melin, 1999). The fractal dimension of a geometrical object can be defined as follows (Peitgen et. al, 1992):

$$d = \lim_{r \to 0} [\ln N(r)] / [\ln(1/r)] \qquad (15.3)$$

where N(r) is the number of structuring elements (boxes, balls, line segments, etc.) covering the object and r is the size of the box (Pesin, 1999). An approximation to the fractal dimension can be obtained by counting the number of boxes covering the boundary of the object for different r sizes and then performing a logarithmic regression to obtain d (box counting algorithm). In Figure 15.3, we illustrate the box counting algorithm for a hypothetical curve C.

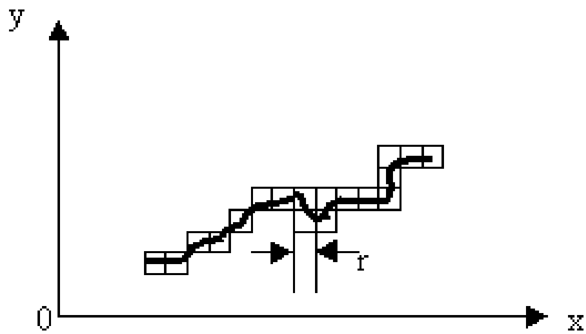


Fig. 15.3. Box Counting Algorithm for a Curve C

Counting the number of boxes for different sizes of r and performing a logarithmic linear regression, we can estimate the box dimension of a geometrical object with the following equation (Melin and Castillo, 2002):

$$\ln N(r) = \ln\beta - d \ln r \tag{15.4}$$

this algorithm is illustrated in Figure 15.4.

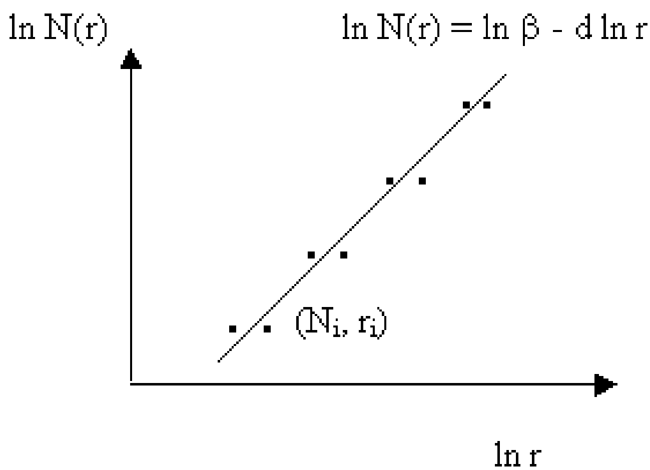


Fig. 15.4. Logarithmic Regression to find the Fractal Dimension

The fractal dimension can be used to characterize an arbitrary object (Tricot, 1995). The reason for this is that the fractal dimension measures the geometrical complexity of objects (Moon, 1992). In this case, a time series can be classified by using the numeric value of the fractal dimension (d is between 1 and 2 because we are on the plane xy).

15.4 Fuzzy Estimation of the Fractal Dimension

The traditional fractal dimension of a geometrical object assigns a crisp numerical value, which measures the geometrical complexity of the object (Semmes, 2000). However, in practice it is difficult to assign a unique numerical value to an object due to uncertainty (Peters, 1994). It is more appropriate to assign a range of numerical values in which there exists a membership degree for this object. For this reason, we will assign to an object O a fuzzy set μ_o, which measures the membership degree for that object. Lets consider, for simplicity, that the object O is in the plane xy, then a suitable membership function could be a generalized bell function (Zadeh 1971):

$$\mu_o = 1 / [\, 1 + |\, (d-c) / a\, |^{2b}\,] \tag{15.5}$$

where a, b and c are the parameters of the membership function and d is the traditional crisp value of the fractal dimension. Of course other types of membership functions could be used depending on the characteristics of the application. By using the concept of a fuzzy set (Zadeh, 1965) we are in fact generalizing the mathematical concept of the fractal dimension because now we can take into account the uncertainties that may arise due to sampling and experimental errors. In fact, our definition of the fuzzy fractal dimension for this case is as follows.

Definition 15.1. Let O be an arbitrary geometrical object in the plane xy. Then the fuzzy fractal dimension is the pair: (d_o, μ_o)

where d_o is the numerical value of the fractal dimension calculated by the box counting algorithm, and μ_o is the membership function for the object.

With this new definition we can account for the uncertainty in the estimation of the fractal dimension of an object. We are, in fact, using this concept of the fuzzy fractal dimension in this paper to consider the uncertainty in the time series analysis required by the monitoring applications. Also, this new definition enables easier pattern recognition for objects, because it is not necessary to match an exact numerical value to recognize a particular object (Yager and Filev, 1994).

15.5 Type-2 Fuzzy Logic

The concept of a type-2 fuzzy set was introduced by Zadeh in 1973 as an extension of the concept of an ordinary fuzzy set (henceforth called a "type-1 fuzzy set"). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership grade for each element of this set is a fuzzy set in [0,1], unlike a type-1 set where the membership grade is a crisp number in [0,1]. Such sets can be used in situations where there is uncertainty about the membership grades themselves, e.g., an uncertainty in the shape of the membership function or in some of its parameters (Mendel, 2001). Consider the transition from ordinary sets to fuzzy sets. When we cannot determine the membership of an element in a set as 0 or 1, we use fuzzy sets of type-1. Similarly, when the situation is so fuzzy that we have trouble determining the membership grade even as a crisp number in [0,1], we use fuzzy sets of type-2.

Example: Consider the case of a fuzzy set characterized by a Gaussian membership function with mean m and a standard deviation that can take values in $[\sigma_1,\sigma_2]$, i.e.,

$$\mu(x)=\exp\{-\tfrac{1}{2}[(x-m)/\sigma]^2\}; \quad \sigma \in [\sigma_1,\sigma_2] \tag{15.6}$$

Corresponding to each value of σ, we will get a different membership curve (see Figure 15.5). So, the membership grade of any particular x (except x=m) can take any of a number of possible values depending upon the value of σ, i.e., the membership grade is not a crisp number, it is a fuzzy set. Figure 15.5 shows the domain of the fuzzy set associated with x=0.7.

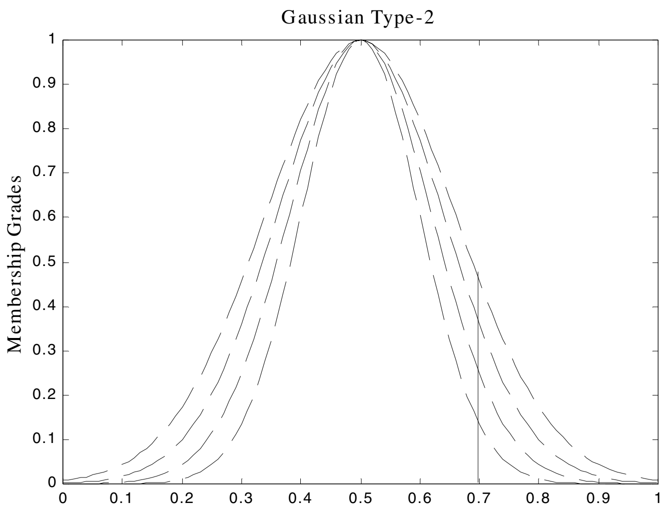


Fig. 15.5. Type-2 fuzzy set representing a type-1 set with uncertain deviation

We can formally define two kinds of type-2 sets as follows.

Definition 2. Gaussian type-2
A Gaussian type-2 fuzzy set is one in which the membership grade of every domain point is a Gaussian type-1 set contained in [0,1].

Definition 3. Interval type-2
An interval type-2 fuzzy set is one in which the membership grade of every domain point is a crisp set whose domain is some interval contained in [0,1].

The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general, will not change for any type-n (Mendel, 2001). A higher-type number just indicates a higher "degree of fuzziness". Since a higher type changes the nature of the membership functions, the operations that depend on the membership functions change; however, the basic principles of fuzzy logic are independent of the nature of

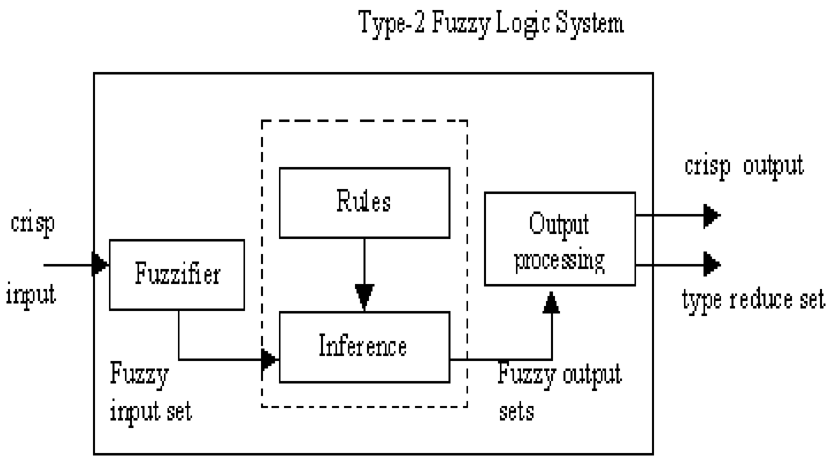


Fig. 15.6. Structure of a type-2 fuzzy system

membership functions and hence, do not change. In Figure 15.6 we show the general structure of a type-2 fuzzy system. We assume that both antecedent and consequent sets are type-2; however, this need not necessarily be the case in practice.

The structure of the type-2 fuzzy rules is the same as for the type-1 case because the distinction between type-2 and type-1 is associated with the nature of the membership functions. Hence, the only difference is that now some or all the sets involved in the rules are of type-2. In a type-1 fuzzy system, where the output sets are type-1 fuzzy sets, we perform defuzzification in order to get a number, which is in some sense a crisp (type-0) representative of the combined output sets. In the type-2 case, the output sets are type-2; so we have to use extended versions of type-1 defuzzification methods. Since type-1 defuzzification gives a crisp number at the output of the fuzzy system, the extended defuzzification operation in the type-2 case gives a type-1 fuzzy set at the output. Since this operation takes us from the type-2 output sets of the fuzzy system to a type-1 set, we can call this operation "type reduction" and call the type-1 fuzzy set so obtained a "type-reduced set". The type-reduced fuzzy set may then be defuzzified to obtain a single crisp number; however, in many applications, the type-reduced set may be more important than a single crisp number.

Type-2 sets can be used to convey the uncertainties in membership functions of type-1 fuzzy sets, due to the dependence of the membership functions on available linguistic and numerical information. Linguistic information (e.g. rules from experts), in general, does not give any information about the shapes of the membership functions. When membership functions are determined or tuned based on numerical data, the uncertainty in the numerical data, e.g., noise, translates into uncertainty in the membership functions. In all such cases, any available information about the linguistic/numerical uncertainty can be incorporated in the type-2 framework. However, even with all of the advantages that fuzzy type-2 systems have, the literature on the applications of type-2 sets is scarce (Mendel, 2001). Some examples are for decision-making (Yager, 1980), and for solving fuzzy relational equations (Wagenknecht and Hartmann, 1988). For the specific case of plant monitoring, the use of type-2 fuzzy rules is justified when the degree of uncertainty is high (for example, due to noise or complexity of the process) in the measured time series. Of course, a type-1 fuzzy system could be enough for plant monitoring in the case of relatively simple processes in the plant (Zadeh, 1975). However, in other cases there are highly non-linear processes present in the plant, like in biochemical reactors or electrochemical processes. For this reason, we are proposing that to model the uncertainty in this type of highly non-linear processes, we need to use type-2 fuzzy logic. We illustrate the application of the type-2 fuzzy logic approach with the case of monitoring the electrochemical process in battery production in a real plant.

15.6 Plant Monitoring Using a Type-1 Fuzzy-Fractal Approach

In this section, we show how to implement a fuzzy rule-based expert monitoring system with two basic sensors: temperature, and pressure. These two physical variables are very important in understanding any chemical process. Our particular case, is the monitoring of an electrochemical process, like the one used in battery formation. We also use as input the fuzzy fractal dimension of the time series of the measured variables. Of course, this fuzzy fractal dimension is not a real physical variable, but we can use it to measure the complexity of the dynamic behavior of the process. We have assigned linguistic values to the fuzzy fractal dimension, with the help of experts in the process in such a way as to help in the diagnostic of the different conditions. Individual sensors can identify three linguistic values (normal, high, and low) for the two real physical variables. The three inputs can be combined to give 9 different real scenarios. Of course, there could be in theory at most 27 scenarios in this case, but there are only 9 real ones for the particular application of the electrochemical process. This is perfectly clear if we notice that there are only two real physical variables with three linguistic values for each one. With the perfectly normal case (where all three input variables have normal values), there are additionally 8 more cases where combinations of abnormal readings can be observed.

Let x_1 be the temperature, x_2 the pressure, x_3 the fuzzy fractal dimension, and y the diagnostic statement. Let L_i, N_i, and H_i, represent the three sets of low range, normal range, and high range for input data x_i, where i = 1, 2, or 3. Furthermore, let C_1, C_2,..., C_9 be the individual scenarios that could happen for each combination of the different data sets. The fuzzy rules have the general form:

$R^{(0)}$: IF x_1 is N_1 AND x_2 is N_2 AND x_3 is N_3 THEN y is C_1

... ...

$R^{(i)}$: IF x_1 is V_1 AND x_2 is V_2 AND x_3 is V_3 THEN y is C_i (15.7)

... ...

$R^{(26)}$: IF x_1 is H_1 AND x_2 is H_2 AND x_3 is H_3 THEN y is C_9

In this case, V_i represents L_i, H_i, or N_i, depending on the condition for the plant. Experts have to provide their knowledge in plant monitoring to label the individual cases Ci for i = 1, 2,..., 9. Also, the membership functions for the linguistic values of variables have to be defined according to historical data of the problem and expert knowledge. Of course, expert knowledge for temperature and pressure is based on the

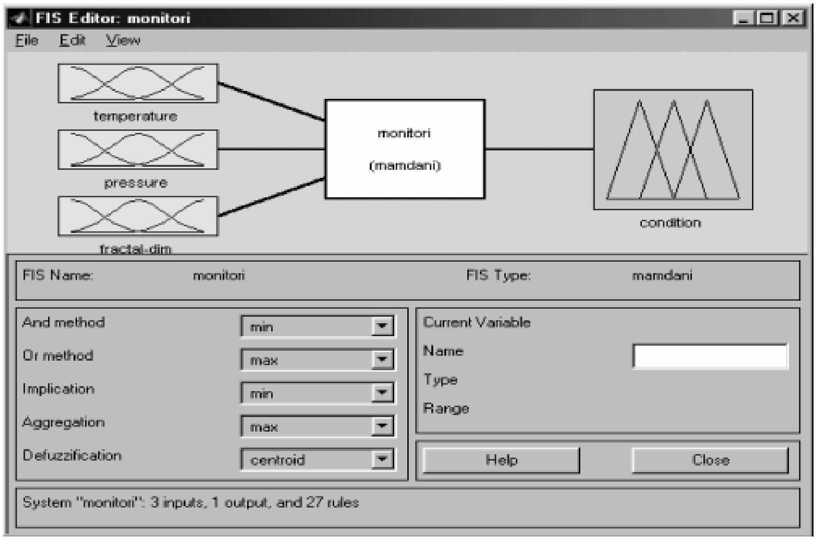


Fig. 15.7. Architecture of the fuzzy system for plant monitoring

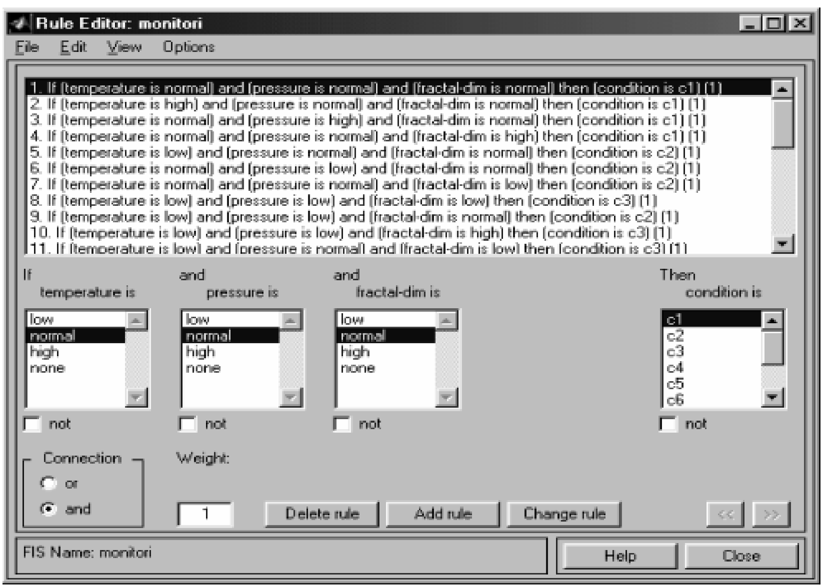


Fig. 15.8. Fuzzy rule base for plant monitoring

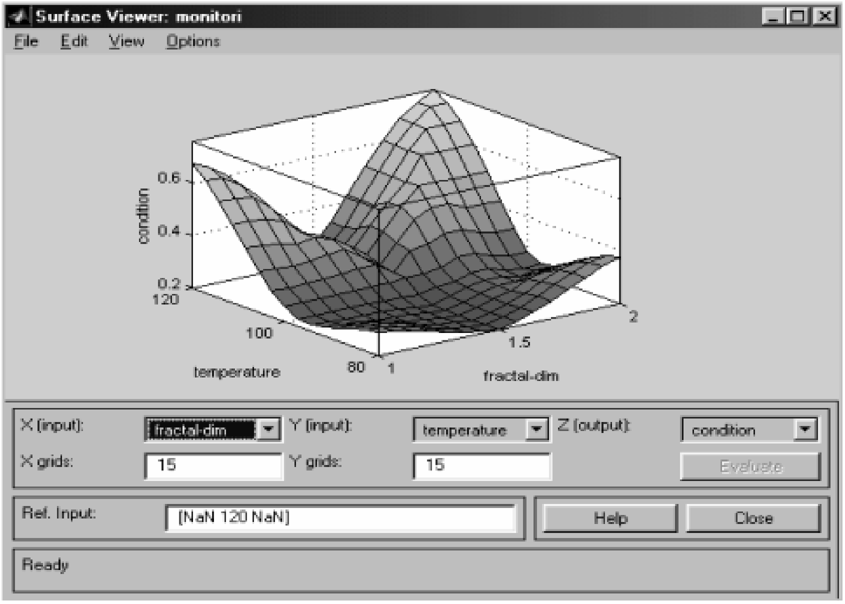


Fig. 15.9. Non-linear surface for plant monitoring with respect to temperature and fractal dimension

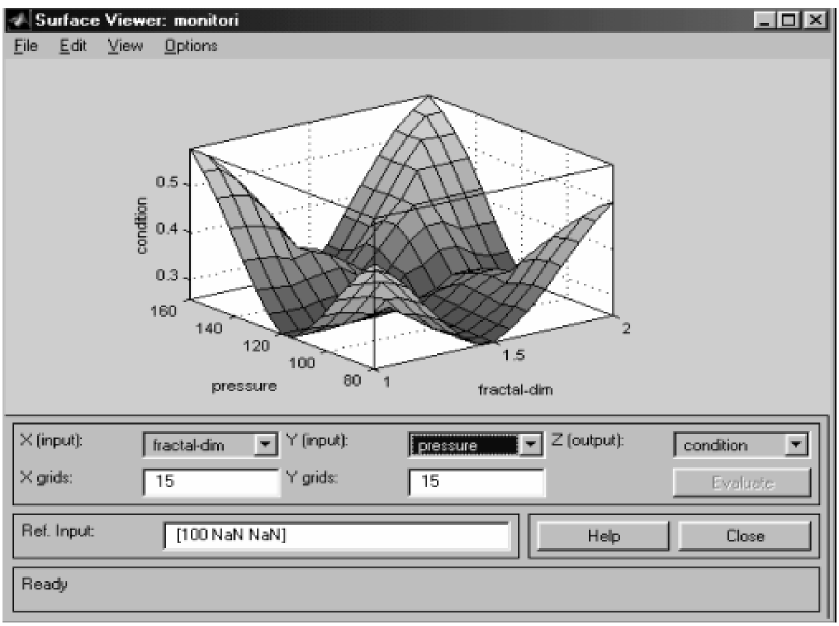


Fig. 15.10. Non-linear surface for plant monitoring with respect to pressure and fractal dimension

dynamics of the process, which experts have gained with their experience. On the other hand, expert knowledge for the fuzzy fractal dimension is more of a combination of expertise on the problem and expertise on the mathematics of fractal theory. Perhaps, this can be explained as follows: we need expert knowledge to know if the dynamics of the process are abnormal or normal, on the other hand we need knowledge on fractal theory to relate the dynamics to a higher or lower value of the fractal dimension.

We can use the Fuzzy Logic Toolbox of the MATLAB language to implement the fuzzy monitoring system described above. In this case, we need to specify the particular fuzzy rules and the corresponding membership functions for the problem. First, we show in Figure 15.7 the general architecture of the fuzzy monitoring system. In this figure, we can see the input linguistic variables (temperature, pressure, and fractal dimension) and the output variable (condition of the plant) of the fuzzy monitoring system. Of course, in this case the fractal dimension is estimated using the box counting algorithm, which was implemented also in MATLAB. In Figure 15.8 the implementation of the fuzzy rule base is shown. The actual 27 rules were defined according to expert knowledge on the process. In Figures 15.9 and 15.10 the non-linear surface for the problem of monitoring is shown.

15.7 Plant Monitoring Using the Type-2 Fuzzy-Fractal Approach

For the case of the type-2 fuzzy-fractal approach, we have to change our way of calculating the output of the fuzzy system. Now, we use interval computations to find the minimum and maximum values for obtaining the outputs of the type-2 fuzzy system. We basically, compute the outputs of two type-1 fuzzy systems, one for computing the minimum value and the other for the maximum value. Then, if we need to further reduce the type of the output, we can apply the traditional deffuzzification methods. Fortunately, in this way we can take advantage of the machinery that we already have for type-1 fuzzy logic, as well as the computer programs in the MATLAB language.

We show below in Table 15.1 the results of a type-2 fuzzy system for monitoring different conditions of the plant. We also show the corresponding values of the type-1 fuzzy logic approach. A comparison, of both approaches can be made in this way. Of course, it is easy to appreciate that in the case of type-2 fuzzy logic the output result is an interval, instead of a single numeric value. In some applications, this is closer to reality, since we are expecting something similar to a confidence interval. However, in other areas of application, like in control, a unique result is needed, so in this case we need to defuzzify again (or type-reduce the result). This is also shown in Table 15.1.

From Table 15.1 we can appreciate that in some cases the output of the type-2 fuzzy system is almost the same as in the type-1 case, but in other situations the results are somewhat different. For our application, we find out that the results of the type-2 fuzzy logic approach were better for monitoring the plant. The main reason for saying that the type-2 approach is better, in this case, is that we are really predicting

Table 15.1. Comparison between the type-2 and type-1 fuzzy logic approaches

Temperature	Pressure	Fractal Dim.	Type-1 F. L.	Type-2 Min	Type-2 Max
105	130	1.6	0.4498	0.4391	0.5030
100	120	1.5	0.2688	0.2774	0.2775
95	115	1.4	0.2263	0.2216	0.2539
90	110	1.3	0.2460	0.2282	0.2783
102	122	1.7	0.3604	0.3321	0.4210
85	90	1.2	0.2690	0.2540	0.2750
75	100	1.8	0.2652	0.2251	0.3039
55	105	1.3	0.2700	0.2700	0.2701
130	90	1.1	0.5710	0.5586	0.5855
112	115	1.6	0.4136	0.4092	0.4138

possible problems in the plant, and the type-2 intervals are closer to what the experts were expecting to see in the monitoring of the process.

We show in Figures 15.11 and 15.12 the non-linear surfaces for the type-1 and type-2 fuzzy systems, respectively. From these figures, we can appreciate the difference between both fuzzy logic approaches. It is obvious that the type-2 fuzzy logic surface is smoother, which is better for modeling the monitoring problem. Finally, we show in Figure 15.13 a comparison between the predicted outputs for the type-2 and type-1 fuzzy systems. In this figure we can appreciate that the type-2 fuzzy logic approach is really modeling our uncertainty about the membership functions of the fuzzy system. For this reason, the type-1 result is almost always in between the minimum and maximum values of the type-2 approach. Of course, the type-2 approach is more realistic because we do not know the exact parameter values of the type-1 membership functions.

Based on the examples presented in this section and the previous one, we see that using fuzzy logic in monitoring and diagnostics always results in improved performance. Also, the use of the fractal dimension improves the accuracy of the method. We have compared the success rate of the type-2 fuzzy-fractal approach, the type-1 fuzzy-fractal approach, and the use of only fuzzy logic, using the data from electrochemical processes in a real plant. The results are shown in Table 15.2. We are using in all of the cases a specific electrochemical process for battery formation. The process is considered to be in a different condition in each of the three cases. The comparison is between the results of the intelligent system using the type-2 fuzzy-fractal approach, the type-1 approach, and a computer program using only fuzzy logic with the Mamdani approach.

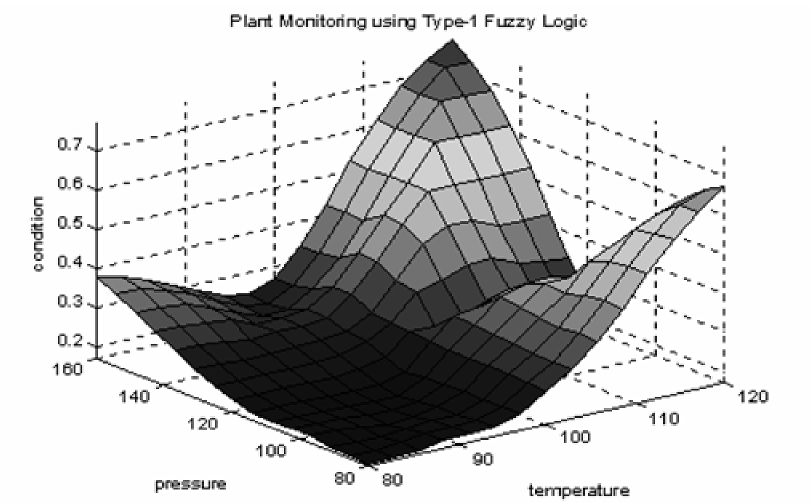


Fig. 15.11. Non-linear surface for the type-1 fuzzy system

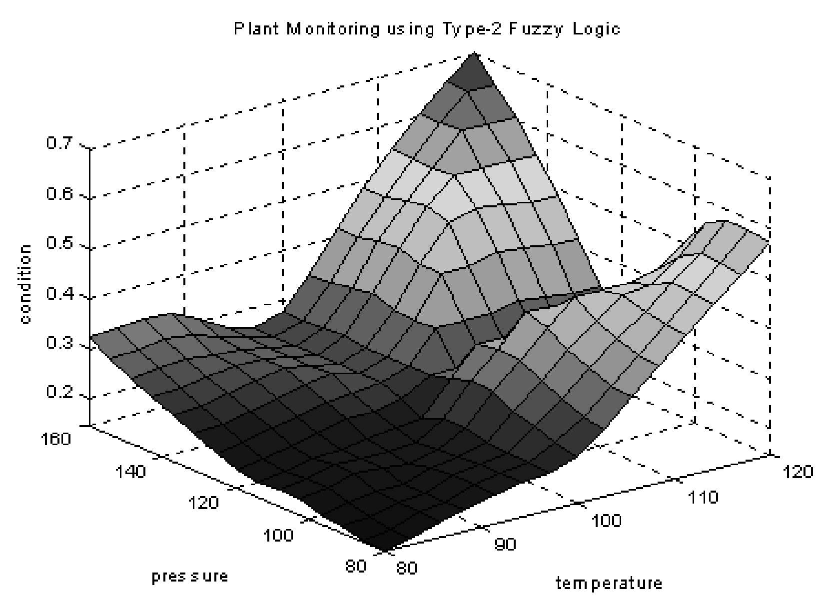


Fig. 15.12. Non-linear surface for the type-2 fuzzy system

It should be pointed out that no matter what techniques are used, there is no guarantee of success because monitoring and diagnosis is a process of abduction. First, the training samples may not represent all the patterns of different system conditions. An

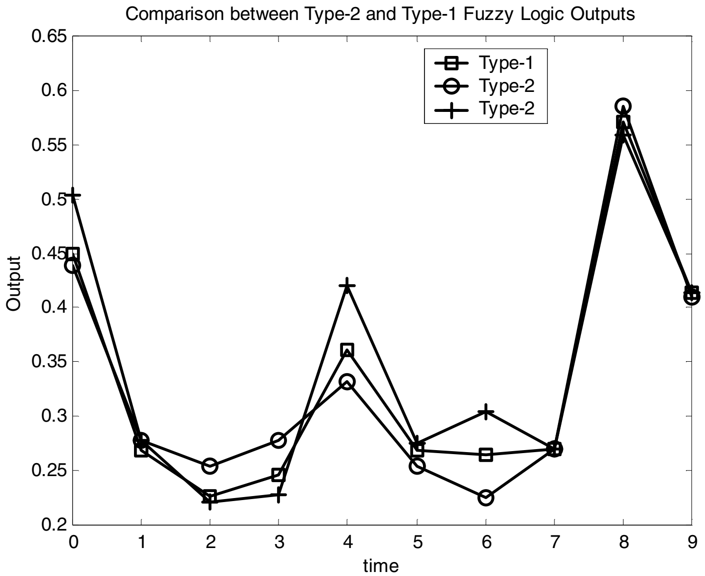


Fig. 15.13. Comparison of outputs of the type-2 and type-1 fuzzy systems

Table 15.2. Success rates of the type-2 and type-1 fuzzy-fractal approaches for monitoring

Applications	Type-2 Fuzzy-Fractal	Type-1 Fuzzy-Fractal	Fuzzy Logic
condition monitoring in an electrochemical process (case 1)	99%	98%	82%
condition monitoring in an electrochemical process (case 2)	88%	86%	73%
condition monitoring in an electrochemical process (case 3)	93%	90%	79%

effective solution to this problem is to add more training samples. Second the patterns of different system conditions overlap and/or are inseparable owing to the definition of system conditions and the use of monitoring indices.

Finally, it is interesting to compare the performance of the fuzzy-fractal approaches with that of using only fuzzy logic (see Table 15.2). We see that the type-1 fuzzy-fractal approach outperforms the fuzzy logic approach by at least 10% in all the cases. We also can appreciate that the type-2 fuzzy-fractal approach outperforms by 2%

(average) the type-1 approach. This demonstrates that the type-2 fuzzy-fractal approach is indeed the more effective method and, in general outperforms the use of type-1 fuzzy logic.

15.8 Summary

In this chapter a hybrid fuzzy-fractal approach for plant monitoring has been proposed. Type-2 fuzzy logic is used to model the uncertainty of plant monitoring and diagnostics. An implementation in MATLAB has been shown, to describe in more detail the advantages of the new approach. The hybrid fuzzy-fractal approach combines the advantages of fuzzy logic (expert knowledge representation) with the advantages of the fractal dimension concept (ability to measure object complexity), to achieve efficient monitoring and diagnostics. A problem yet to be considered, is how to automatically learn (or adapt) the membership functions and rules of the fuzzy system using real data for the problem.

16 Intelligent Control of Autonomous Robotic Systems Using Interval Type-2 Fuzzy Logic and Genetic Algorithms

We develop a tracking controller for the dynamic model of unicycle mobile robot by integrating a kinematic controller and a torque controller based on Fuzzy Logic Theory. Computer simulations are presented confirming the performance of the tracking controller and its application to different navigation problems.

16.1 Introduction

Mobile robots are nonholonomic systems due to the constraints imposed on their kinematics. The equations describing the constraints cannot be integrated simbolically to obtain explicit relationships between robot positions in local and global coordinate's frames. Hence, control problems involve them have attracted attention in the control community in the last years (Kolmanovsky and McClamroch, 1995).

Different methods have been applied to solve motion control problems. (Kanayama et al., 1991) propose a stable tracking control method for a nonholonomic vehicle using a Lyapunov function. (Lee et al., 1998) solved tracking control using backstepping and in (Lee and Tai, 2001) with saturation constraints. Furthermore, most reported designs rely on intelligent control approaches such as Fuzzy Logic Control (Bentalba et al., 1997) (Ishikawa, 1991) (Lee et al., 1999) (Pawlowski, 2001) and Neural Networks (Fierro and Lewis, 1998) (Song and Sheen, 2000).

However the majority of the publications mentioned above, has concentrated on kinematics models of mobile robots, which are controlled by the velocity input, while less attention has been paid to the control problems of nonholonomic dynamic systems, where forces and torques are the true inputs: (Bloch and Drakunov, 1991) and (Chwa, 2004), used a sliding mode control to the tracking control problem. (Fierro and Lewis, 1995) propose a dynamical extension that makes possible the integration of kinematic and torque controller for a nonholonomic mobile robot. (Fukao et al., 2000), introduced an adaptive tracking controller for the dynamic model of mobile robot with unknown parameters using backstepping.

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 203–212, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

In this chapter we present a tracking controller for the dynamic model of a unicycle mobile robot, using a control law such that the mobile robot velocities reach the given velocity inputs, and a fuzzy logic controller such that provided the required torques for the actual mobile robot. The rest of this chapter is organized as follows. Sections 16.2 and 16.3 describe the formulation problem, which include: the kinematic and dynamic model of the unicycle mobile robot and introduces the tracking controller. Section 16.4 illustrates the simulation results using the tracking controller. The section 16.5 gives the conclusions.

16.2 Problem Formulation

The model considered in this chapter is of a unicycle mobile robot (see Figure 16.1), it consist of two driving wheels mounted on the same axis and a front free wheel (Campion et al., 1996).

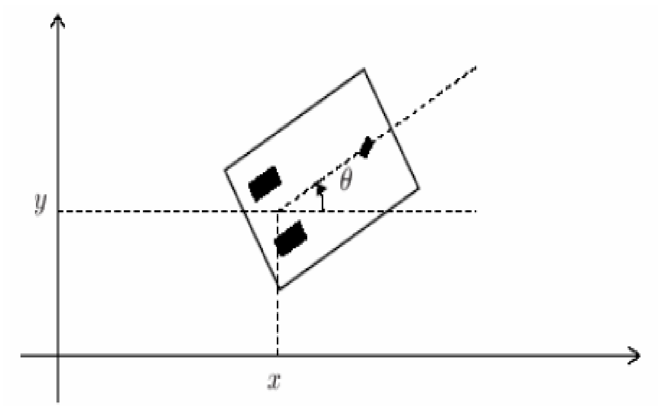


Fig. 16.1. Wheeled mobile robot

The motion can be described with equation (16.1) of movement in a plane (Fierro and Lewis, 1995):

$$\dot{q} = \begin{vmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} v \\ w \end{vmatrix}$$

$$M(q)\dot{v} + V(q,\dot{q})v + G(q) = \tau \tag{16.1}$$

Where $q = [x, y, \theta]^T$ is the vector of generalized coordinates which describes the robot position, *(x,y)* are the cartesian coordinates, which denote the mobile center of mass and θ is the angle between the heading direction and the *x*-axis (which is taken counterclockwise form); $v = [v, w]^T$ is the vector of velocities, *v* and *w* are the linear and angular velocities respectively; $\tau \in R^r$ is the input vector, $M(q) \in R^{nxn}$ is a symmetric and positive-definite inertia matrix, $V(q,\dot{q}) \in R^{nxn}$ is the centripetal and Coriolis matrix, $G(q) \in R^n$ is the gravitational vector. Equation (16.1) represents the kinematics or steering system of a mobile robot. Notice that the no-slip condition imposed a non-holonomic constraint described by equation (16.2), that it means that the mobile robot can only move in the direction normal to the axis of the driving wheels.

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{16.2}$$

Tracking Controller of Mobile Robot

Our control objective is established as follows: Given a desired trajectory $q_d(t)$ and orientation of mobile robot we must design a controller that apply adequate torque τ such that the measured positions $q(t)$ achieve the desired reference $q_d(t)$ represented as equation (16.3):

$$\lim_{t\to\infty}\left\|q_d(t) - q(t)\right\| = 0 \tag{16.3}$$

To reach the control objective, we are based in the procedure of (Fierro and Lewis, 1995), we deriving a $\tau(t)$ of a specific $v_c(t)$ that controls the steering system (16.1) using a Fuzzy Logic Controller (FLC). A general structure of tracking control system is presented in the Figure 16.2.

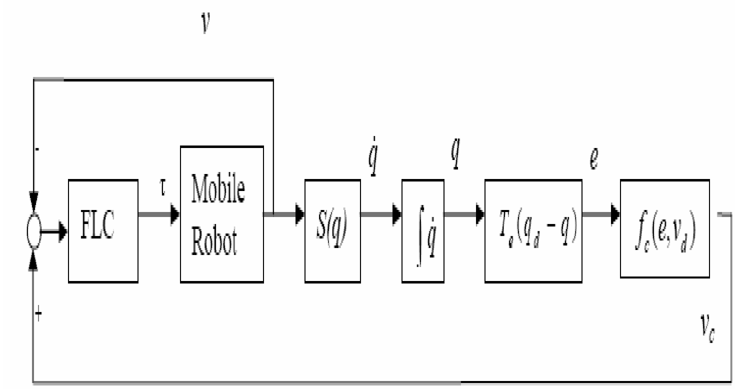


Fig. 16.2. Tracking control structure

16.3 Control of the Kinematic Model

We are based on the procedure proposed by (Kanayama et al., 1991) and (Nelson et al., 1988) to solve the tracking problem for the kinematic model, this is denoted as $v_c(t)$. Suppose the desired trajectory q_d satisfies equation (16.4):

$$\dot{q}_d = \begin{vmatrix} \cos\theta_d & 0 \\ \sin\theta_d & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} v_d \\ w_d \end{vmatrix} \tag{16.4}$$

Using the robot local frame (the moving coordinate system x-y in figure 16.1), the error coordinates can be defined as equation (16.5):

$$e = T_e(q_d - q), \begin{vmatrix} e_x \\ e_y \\ e_\theta \end{vmatrix} = \begin{vmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{vmatrix} \tag{16.5}$$

And the auxiliary velocity control input that achieves tracking for equation (16.1) is given by equation (16.6):

$$v_c = f_c(e, v_d), \begin{vmatrix} v_c \\ w_c \end{vmatrix} = \begin{vmatrix} v_d + \cos e_\theta + k_1 e_x \\ w_d + v_d k_2 e_y + v_d k_3 \sin e_\theta \end{vmatrix} \tag{16.6}$$

Where k_1, k_2 and k_3 are positive constants.

16.4 Fuzzy Logic Controller

The purpose of the Fuzzy Logic Controller (FLC) is to find a control input τ such that the current velocity vector v to reach the velocity vector v_c this is denoted in equation (16.7):

$$\lim_{t \to \infty} \left\| v_c - v \right\| = 0 \tag{16.7}$$

As is shown in Figure 16.2, basically the FLC has 2 input variables corresponding the velocity errors obtained of (16.7) (denoted as ev and ew: linear and angular velocity errors respectively), and 2 outputs variables, the driving and rotational input torques τ (denoted by F and N respectively). The membership functions (MF)[9] are defined by 1 triangular and 2 trapezoidal functions for each variable involved due to the fact are easy to implement computationally.

Figures 16.3 and Fig. 16.4 show the MFs in which N, C, P represent the fuzzy sets [9] (Negative, Zero and Positive respectively) associated to each input and output variable, where the universe of discourse is normalized into [-1,1] range.

The rule set of FLC contain 9 rules which governing the input-output relationship of the FLC and this adopts the Mamdani-style inference engine (Passino andYurkovich, 1998), and we use the center of gravity method to realize defuzzification procedure. In Table 16.1, we present the rule set whose format is established as follows:

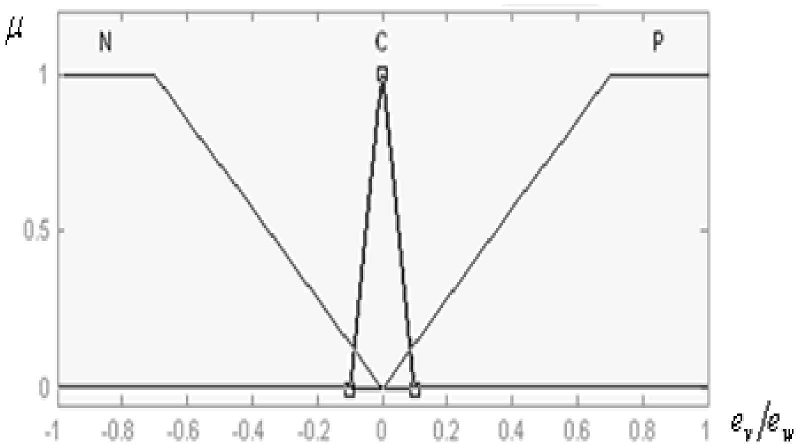


Fig. 16.3. Membership function of the input variables e_v and e_w

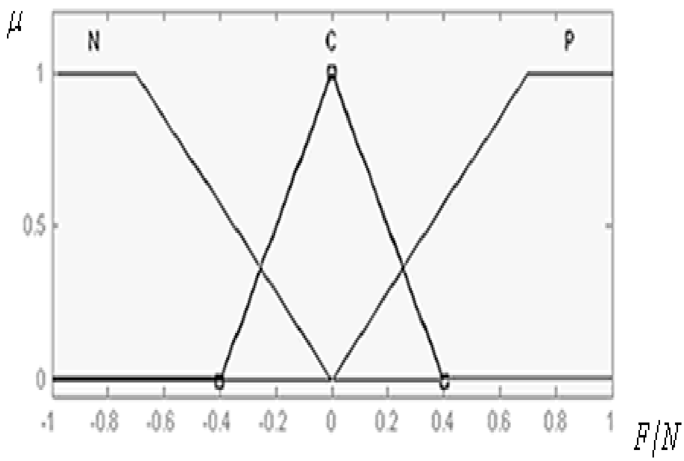


Fig. 16.4. Membership functions of the output variables F and N

Table 16.1. Fuzzy rule set

e_v / e_w	N	C	P
N	*N/N*	*N/C*	*N/P*
C	*C/N*	*C/C*	*C/P*
P	*P/N*	*P/C*	*P/P*

Rule *i*: If e_v is G_1 and e_w is G_2 then F is G_3 and N is G_4

Where G_1..G_4 are the fuzzy set associated to each variable and *i*= 1 ... 9. In Table 16.1, N means NEGATIVE, P means POSITIVE and C means ZERO.

16.5 Simulation Results

Simulations have been done in Matlab® to test the tracking controller of the mobile robot defined in equation (16.1). We consider the initial position $q(0) = (0, 0, 0)$ and initial velocity $v(0) = (0,0)$. From Figures 16.5 to 16.8 we show the results of the simulation for the case 1. Position and orientation errors are depicted in Figure 16.5 and 16.6 respectively, as can be observed the errors are sufficient close to zero, the trajectory tracked (see Figure 16.7) is very close to the desired, and the velocity errors shown in Figure 16.8 decrease to zero, achieving the control objective in less than 1 second of the whole simulation. We show in Figure 16.9 the Simulink block diagram to test the controller. We also show in Figure 16.10 the tracking errors in the three variables. Finally, we show in Figure 16.11 the evolution of the genetic algorithm that was used to find the optimal parameters for the fuzzy controller.

In Table 16.2 we show simulation results for 25 experiments with different conditions for the gains of the fuzzy controller. We can also appreciate from this table that different reference velocities and positions were considered.

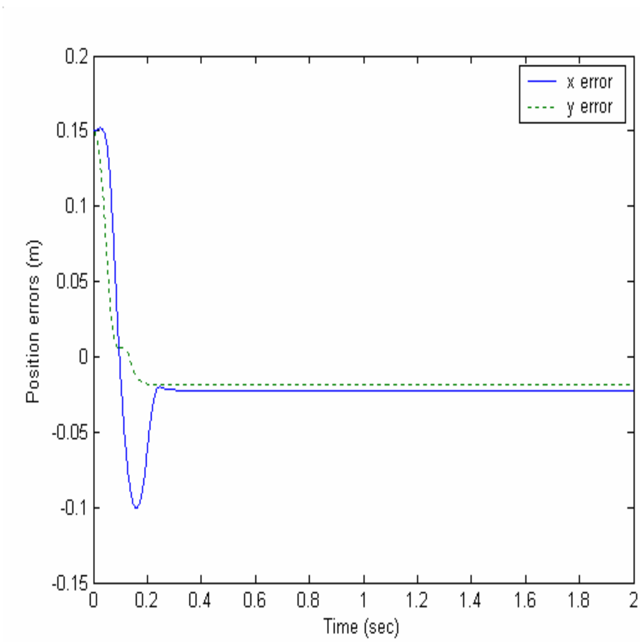


Fig. 16.5. Positions error with respect to the reference values. Solid: error in x, dotted: error in y.

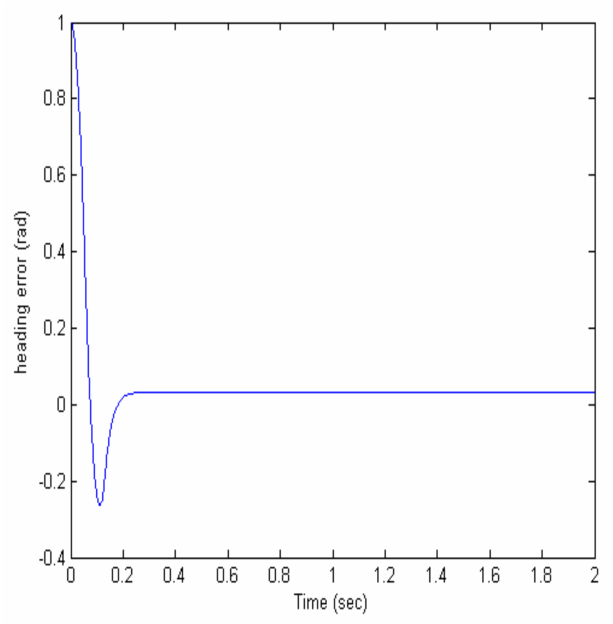


Fig. 16.6. Orientation error with respect to the reference values

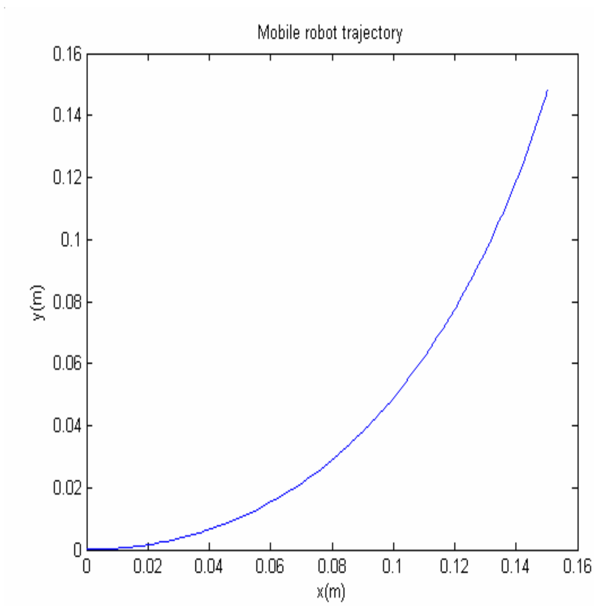


Fig. 16.7. Mobile Robot Trajectory

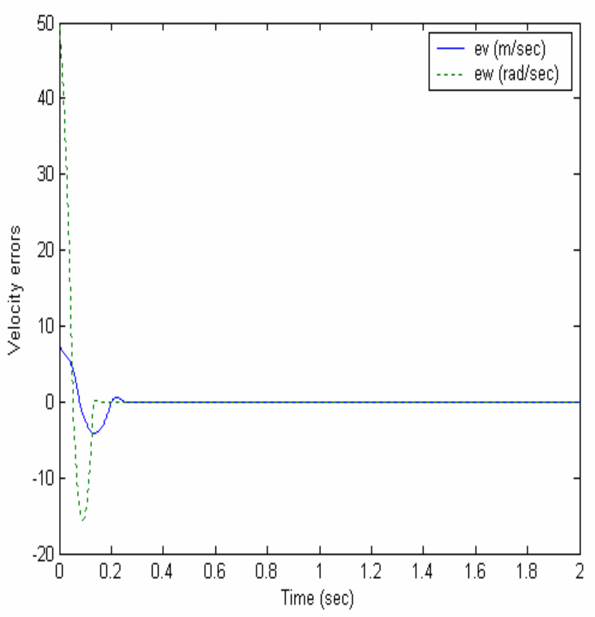


Fig. 16.8. Velocity errors: Solid: error in e_v, dotted: error in e_w

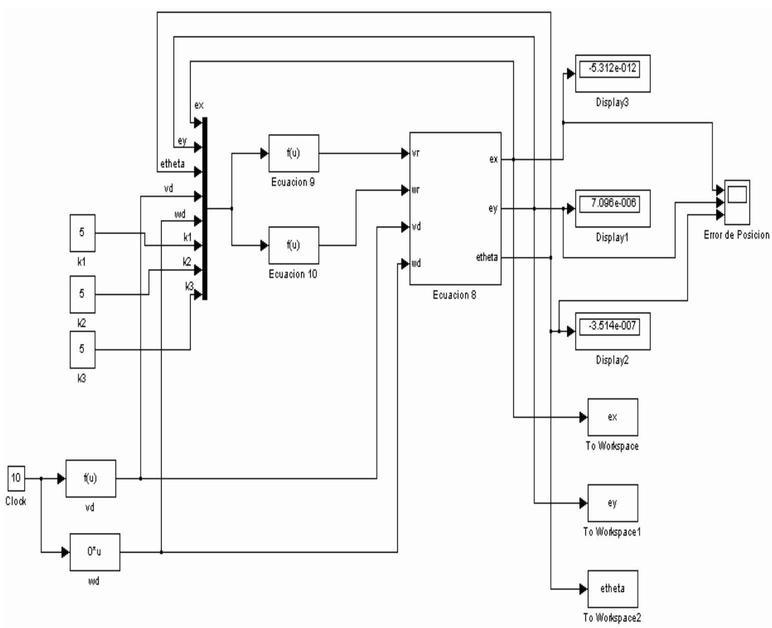


Fig. 16.9. Simulink block diagram of the controller

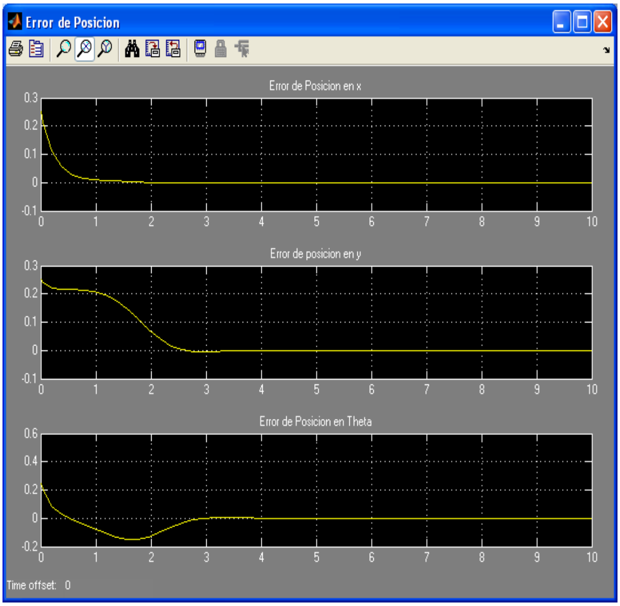


Fig. 16.10 Tracking errors in the three variables

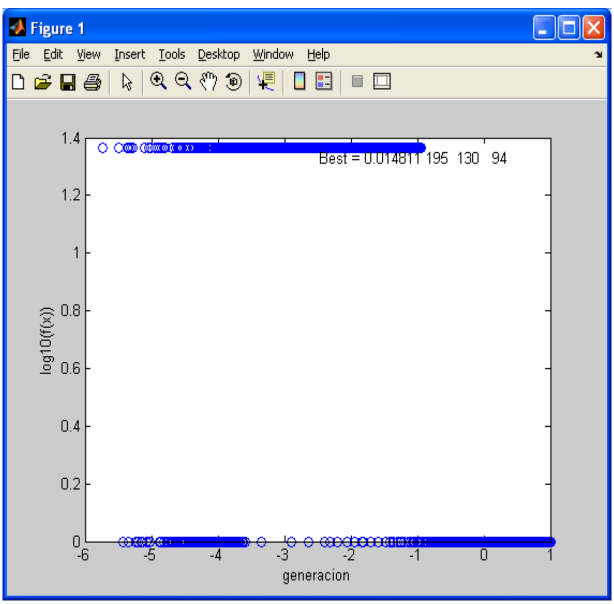


Fig. 16.11 Evolution of GA for finding optimal Controller

Table 16.2. Simulation results for different experiments with the fuzzy controller

No.	Controlador	Escala	Velocidades Iniciales		e					Constantes		
			vd	wd	ex	ey	etheta	ev	ew	k1	k2	k3
1	4	[-100 100]	$0.25-0.25\cos(2\pi t/5)$	0	-0.435	2.36	2.79	0.0139	4.4	5	5	5
2	4	[-100 100]	$0.25-0.25\cos(2\pi t/5)$	0	-0.339	-0.0997	-0.5	-14	-0.0773	45	70	45
3	4	[-100 100]	$0.25-0.25\cos(2\pi t/5)$	0	-0.339	-0.0996	-0.5	-32.9	-0.0311	100	160	110
4	4	[-100 100]	0.1	0	-0.323	-0.106	-4.491	0.232	0.4	5	5	5
5	4	[-100 100]	0.1	0	-0.323	-0.106	-0.491	0.232	0.4	45	70	45
6	4	[-100 100]	0.1	0	-0.329	-0.104	-0.498	-30	0.328	100	160	110
7	4	[-100 100]	$0.15-0.15\cos(2\pi t/5)$	0	-0.339	-0.0997	-0.5	-0.6	0.0859	5	5	5
8	4	[-100 100]	$0.15-0.15\cos(2\pi t/5)$	0	-0.314	-0.112	-0.494	-10	0.627	45	70	45
9	4	[-100 100]	$0.15-0.15\cos(2\pi t/5)$	0	-0.314	-0.112	-0.494	-10	0.627	100	160	110
10	4	[-100 100]	0.5	0	1.24	1.16	1.25	5.33	5.32	5	5	5
11	4	[-100 100]	0.5	0	-0.316	-0.108	-0.484	-10.5	-11.8	45	70	45
12	4	[-100 100]	0.5	0	-0.336	-0.101	-0.449	-31.5	-33.6	100	160	110
13	5	[-100 100]	$0.25-0.25\cos(2\pi t/5)$	0	-0.339	-0.0995	-0.5	-0.818	-0.000181	5	5	5
14	5	[-100 100]	$0.25-0.25\cos(2\pi t/5)$	0	-0.339	-0.0996	-0.5	-14.2	0.0325	45	70	45
15	5	[-100 100]	$0.25-0.25\cos(2\pi t/5)$	0	-0.339	-0.0995	-0.5	-33	0.00623	100	160	110
16	5	[-100 100]	0.1	0	-0.333	-0.102	-0.496	-0.185	0.185	5	5	5
17	5	[-100 100]	0.1	0	1.5	-0.425	1.81	40.9	8.57	45	70	45
18	5	[-100 100]	0.1	0	-0.338	-0.1	-0.499	-32.1	-6.57	100	160	110
19	5	[-100 100]	0.5	0	-0.285	-0.119	-0.466	1.21	-1.29E-14	5	5	5
20	5	[-100 100]	0.5	0	-0.3	-0.113	-0.472	-9	-11	45	70	45
21	5	[-100 100]	0.5	0	-0.318	-0.107	-0.485	-28	-31.9	100	160	110
13	1	[-100 100]	$0.25-0.25\cos(2\pi t/5)$	0	-0.319	-0.106	-0.482	0.311	1.01	5	5	5
14	1	[-100 100]	$0.25-0.25\cos(2\pi t/5)$	0	-0.327	-0.104	-0.491	-12.9	0.694	45	70	45
15	1	[-100 100]	$0.25-0.25\cos(2\pi t/5)$	0	-0.339	-0.0997	-0.5	-32.8	0.0839	100	160	110
16	1	[-100 100]	0.1	0	-0.241	-0.127	-0.419	2	1.84	5	5	5
17	1	[-100 100]	0.1	0	-0.312	-0.109	-0.483	-11.8	-1.87	45	70	45
18	1	[-100 100]	0.1	0	-0.321	-0.108	-0.494	-29.3	-6.32	100	160	110
19	1	[-100 100]	0.5	0	0.0567	-0.17	-0.226	6	2.76	5	5	5
20	1	[-100 100]	0.5	0	-0.309	-0.113	-0.487	-11.4	-13.6	45	70	45
21	1	[-100 100]	0.5	0	-0.29	-0.118	-0.473	-25	-32.5	100	160	110

16.6 Summary

We described the development of a tracking controller integrating a fuzzy logic controller for a unicycle mobile robot with known dynamics, which can be applied for both, point stabilization and trajectory tracking. Computer simulation results confirm that the controller can achieve our objective. As future work, several extensions can be made to the control structure of Figure 16.2, such as to increase the tracking accuracy and the performance level.

17 Adaptive Noise Cancellation Using Type-2 Fuzzy Logic and Neural Networks

We describe in this chapter the application of type-2 fuzzy logic for achieving adaptive noise cancellation. The objective of adaptive noise cancellation is to filter out an interference component by identifying a model between a measurable noise source and the corresponding un-measurable interference. In this chapter, we propose the use of type-2 fuzzy logic to find this model. The use of type-2 fuzzy logic is justified due to the high level of uncertainty of the process, which makes difficult to find appropriate parameter values for the membership functions.

17.1 Introduction

Adaptive noise cancellation was first proposed by (Widrow and Glover in 1975); the objective is to filter out an interference component by identifying a linear model between a measurable noise source and the corresponding un-measurable interference. Adaptive noise cancellation using linear filters has been used successfully in real-world applications, such as interference cancellation in electrocardiograms (ECGs), echo elimination on long-distance telephone transmission lines, and antenna interference canceling (Widrow and Stearns, 1985).

It is obvious that we can extend the concept of linear adaptive noise cancellation into the non-linear arena by using non-linear adaptive systems. In this chapter, we show how a neural network can be used to adapt the parameters of a type-2 fuzzy system, which models the dynamics that transforms a noise source into an interference component in a detected signal. Under certain conditions, the proposed approach is sometimes more suitable than noise elimination techniques based on frequency-selective filtering.

Figure 17.1 shows the schematic diagram of an ideal situation to which adaptive noise cancellation can be applied. Here we have an un-measurable information signal x(k) and a measurable noise source signal n(k); the noise source goes through unknown non-linear dynamics (Rasband, 1990) to generate a distorted noise d(k), which

O. Castillo and P. Melin: Type-2 Fuzzy Log.: Theory and Appl., STUDFUZZ 223, pp. 213–223, 2008.
springerlink.com Springer-Verlag Berlin Heidelberg 2008

is then added to x(k) to form the measurable output signal y(k). Our task is to retrieve the original information signal x(k) from the overall output signal y(k), which consists of the information signal x(k) plus d(k), a distorted and delayed version of n(k).

A good example of noise cancellation is the suppression of the maternal ECG component (noise) from the fetal ECG (Jang et al., 1997). Suppose that we want to measure the fetal ECG x(k) during labor. If we record signals from a sensor placed in the abdominal region, the obtained signal is inevitable noisy due to the mother's heartbeat signal n(k), which can be measured clearly via a sensor at the thoracic region. However, the heartbeat signal n(k) does not appear directly in y(k). Instead, n(k) travels through the mother's body and arrives delayed and distorted to appear in the overall measurement y(k). Mathematically, the detected output signal can be expressed as

$$y(k) = x(k) + d(k) = x(k) + f(n(k), n(k-1), n(k-2),\ldots) \tag{17.1}$$

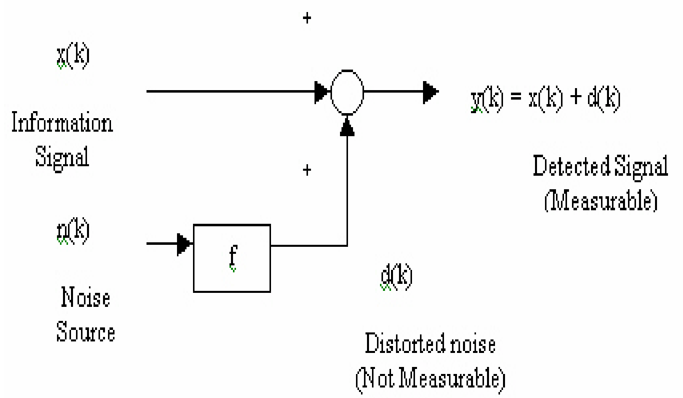


Fig. 17.1. Schematic diagram of noise cancellation

The function f(.) represents the non-linear dynamics that the noise signal n(k) goes through. If f(.) was know exactly, it would be easy to recover the original information signal by subtracting d(k) from y(k) directly. However, f(.) is usually unknown in advance and could be time varying due to changes in the environment. Moreover, the spectrum of d(k) may overlap that of x(k) substantially, invalidating the use of common frequency-domain filtering techniques.

To estimate the distorted noise signal d(k), we need to pick up a clean version of the noise signal n(k) that is independent of the information signal. However, we can not access the distorted noise signal d(k) directly since it is an additive component of the overall measurable signal y(k). Fortunately, as long as the information signal x(k) is zero mean and not correlated with the noise signal n(k), we can use the detected signal y(k) as the desired output of the fuzzy system (trained with a neural network), as in Figure 17.2.

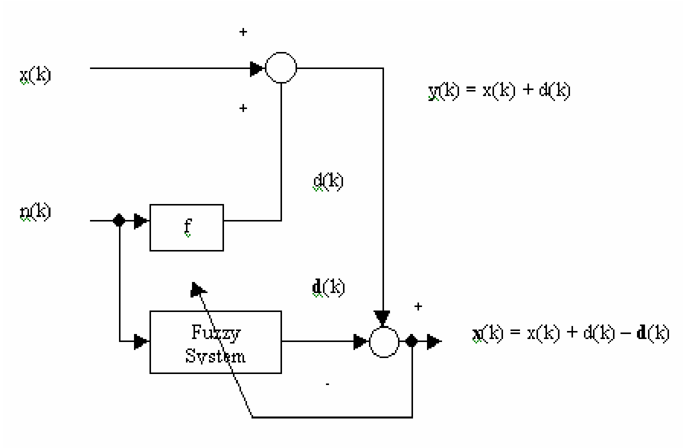


Fig. 17.2. Schematic diagram of noise cancellation with adaptive fuzzy filtering

More specifically, let the output of the fuzzy system be denoted by **d(k)**. The learning rule of the neural network tries to minimize the error

$$\begin{aligned}[e(k)]^2 &= [y(k) - \mathbf{d(k)}]^2 \qquad (17.2)\\ &= [x(k) + d(k) - \mathbf{d(k)}]^2 \\ &= [x(k) + d(k) - \mathbf{f}(n(k), n(k\text{-}1), n(k\text{-}2),\ldots)]^2\end{aligned}$$

where f is the function implemented by the fuzzy system. Since x(k) is not correlated with n(k) or its history, the fuzzy system has no information on how to minimize the error component attributable to x. In other words, the information signal x serves as an un-correlated "noise" component in the data fitting processing. Of course, the fuzzy system can be of conventional type-1 form or of a higher order form, like in the case of a type-2 approximation. We will discuss briefly type-2 fuzzy systems in the following section, assuming that the type-1 case is well known.

17.2 Type-2 Fuzzy Logic

The concept of a type-2 fuzzy set, was introduced by (Zadeh, 1975) as an extension of the concept of an ordinary fuzzy set (henceforth called a “type-1 fuzzy set”). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership grade for each element of this set is a fuzzy set in [0,1], unlike a type-1 set where the membership grade is a crisp number in [0,1]. Such sets can be used in situations where there is uncertainty about the membership grades themselves, e.g., an uncertainty in the shape of the membership function or in some of its parameters. Consider the transition from ordinary sets to fuzzy sets. When we cannot determine the membership of an element in a set as 0 or 1, we use fuzzy sets of type-1. Similarly, when the situation is so fuzzy that we have trouble determining the membership grade even as a crisp number in [0,1], we use fuzzy sets of type-2 (Karnik and Mendel, 1998).

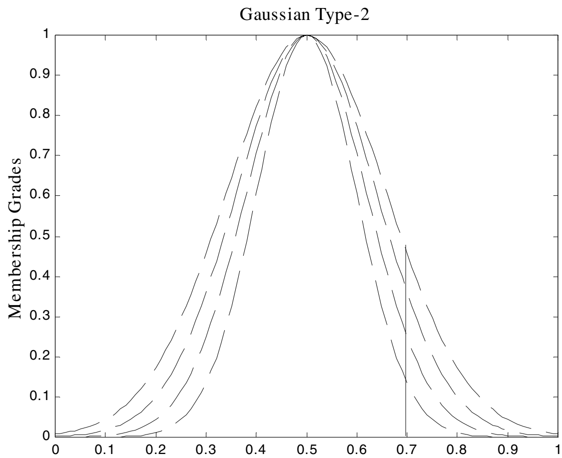


Fig. 17.3. A type-2 fuzzy set representing a type-1 set with uncertain deviation

Example: Consider the case of a fuzzy set characterized by a Gaussian membership function with mean m and a standard deviation that can take values in $[\sigma_1,\sigma_2]$, i.e.,

$$\mu(x)=\exp\{-½[(x-m)/\sigma]^2\}; \quad \sigma \in [\sigma_1,\sigma_2] \tag{17.3}$$

Corresponding to each value of σ, we will get a different membership curve (see Figure 17.3). So, the membership grade of any particular x (except x=m) can take any of a number of possible values depending upon the value of σ, i.e., the membership grade is not a crisp number, it is a fuzzy set. Figure 17.3 shows the domain of the fuzzy set associated with x=0.7.

We can formally define two kinds of type-2 sets as follows.

Definition 17.1. Gaussian type-2
A Gaussian type-2 fuzzy set is one in which the membership grade of every domain point is a Gaussian type-1 set contained in [0,1].

Definition 17.2. Interval type-2
An interval type-2 fuzzy set is one in which the membership grade of every domain point is a crisp set whose domain is some interval contained in [0,1].

The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general, will not change for any type-n (Mendel, 2001). A higher-type number just indicates a higher "degree of fuzziness". Since a higher type changes the nature of the membership functions, the operations that depend on the membership functions change; however, the basic principles of fuzzy logic are independent of the nature of membership functions and hence, do not change. In Figure 17.4 we show the general structure of a type-2 fuzzy system. We assume that both antecedent and consequent sets are type-2; however, this need not necessarily be the case in practice.

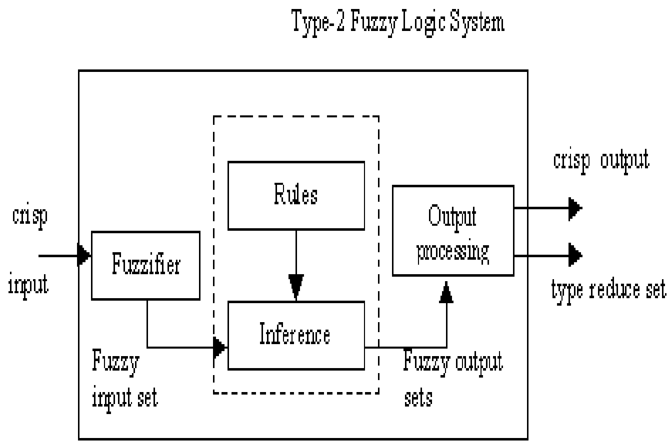


Fig. 17.4. Structure of a type-2 fuzzy system

The structure of the type-2 fuzzy rules is the same as for the type-1 case because the distinction between type-2 and type-1 is associated with the nature of the membership functions. Hence, the only difference is that now some or all the sets involved in the rules are of type-2. In a type-1 fuzzy system, where the output sets are type-1 fuzzy sets, we perform defuzzification in order to get a number, which is in some sense a crisp (type-0) representative of the combined output sets. In the type-2 case, the output sets are type-2; so we have to use extended versions of type-1 defuzzification methods. Since type-1 defuzzification gives a crisp number at the output of the fuzzy system, the extended defuzzification operation in the type-2 case gives a type-1 fuzzy set at the output. Since this operation takes us from the type-2 output sets of the fuzzy system to a type-1 set, we can call this operation "type reduction" and call the type-1 fuzzy set so obtained a "type-reduced set". The type-reduced fuzzy set may then be defuzzified to obtain a single crisp number; however, in many applications, the type-reduced set may be more important than a single crisp number.

Type-2 sets can be used to convey the uncertainties in membership functions of type-1 fuzzy sets, due to the dependence of the membership functions on available linguistic and numerical information. Linguistic information (e.g. rules from experts), in general, does not give any information about the shapes of the membership functions. When membership functions are determined or tuned based on numerical data, the uncertainty in the numerical data, e.g., noise, translates into uncertainty in the membership functions. In all such cases, any available information about the linguistic/numerical uncertainty can be incorporated in the type-2 framework. However, even with all of the advantages that fuzzy type-2 systems have, the literature on the applications of type-2 sets is scarce. Some examples are for decision-making (Yager, 1980), and for solving fuzzy relational equations (Wagenknecht, and Hartmann, 1988). For the specific case of pattern recognition, the use of type-2 fuzzy rules is justified when the degree of uncertainty is high (for example, due to noise or complexity of the process) in the measured time series.

17.3 ANFIS Approach for Building a Type-1 Fuzzy System for Noise Cancellation

ANFIS stands for adaptive neuro fuzzy inference systems (Jang et al., 1997). This architecture was proposed by Jang in 1997 to give fuzzy systems adaptive capabilities. In this way, one can use data to find the optimal parameter values of the fuzzy system. Of course, this ANFIS architecture was initially proposed for the type-1 fuzzy system case. We describe in this section how can the ANFIS approach be used to achieve noise cancellation with a type-1 fuzzy system.

In this case, function **f** of Equation (17.1) is approximated by the fuzzy system generated using ANFIS. Since, the signal x(k) is not correlated with the noise n(k) or its history, ANFIS has no clue on how to minimize the error component attributable to x. In other words, the information signal x serves as an un-correlated "noise" component in the data fitting process, so ANFIS can do nothing about it except picking up its steady-state trend. Instead, the best that ANFIS can do is to minimize the error component attributable to d(k) and this happens to be the desired error measure.

Before presenting simulation results, we establish the conditions under which adaptive noise cancellation is valid:

1) The noise signal n(k) should be available and independent of the information signal x(k).
2) The information signal x(k) must be zero mean.
3) The order of the passage dynamics is known. (This determines the number of inputs to the ANFIS filter.)

In our experiments, we applied ANFIS to two non-linear passage dynamics of orders two and three, respectively. In the first, experiment, the unknown non-linear passage dynamics were assumed to be defined as

$$\begin{aligned} d(k) &= f(n(k), n(k-1)) \\ &= \sin(n(k))\, n(k-1)/ [\ 1 + (n(k-1))^2 \end{aligned} \tag{17.4}$$

where n(k) is a noise source and d(k) denotes the resultant from the non-linear passage dynamics f(.) attributable to n(k) and n(k-1).

We assume that the information signal x(k) is expressed as given in the following equation

$$x(k) = \sin\,(7800/\,k + 20) \tag{17.5}$$

where k is a step count and the sampling periods is of 0.000005 seconds.

To use ANFIS in this experiment, we collected 600 training data pairs of the following form

$$[n(k), n(k-1); y(k)] \tag{17.6}$$

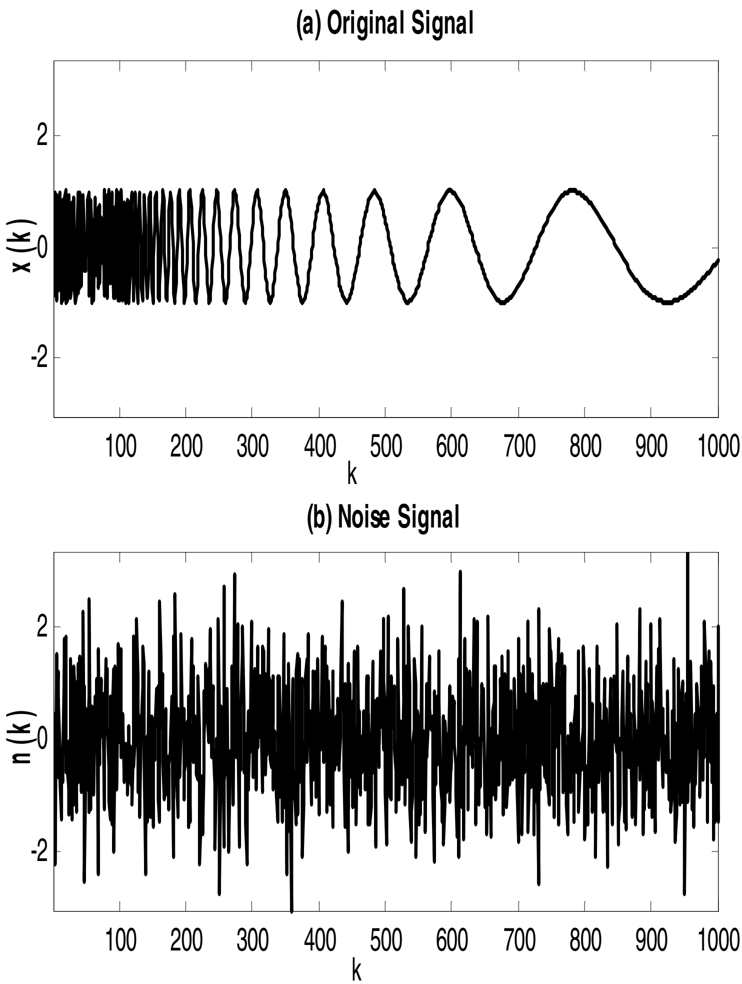


Fig. 17.5. (a) Original information signal x(k), (b) noise signal n(k)

with k runs from 1 to 600. We used a four-rule ANFIS to fit the training data, in which each of the two inputs was assigned two generalized bell membership functions.

In our second experiment, we used real-world audio signals for the simulations. The audio signals were obtained from the MATLAB sound files: *handel.mat* and *chirp.mat*. When these files are loaded into MATLAB and played by the command *sound, handel.mat* is a piece of music of composer George Handel's and *chirp.mat* the *sound of a bird's chirping. In this case, we used handel*.mat as the information signal x(k) and *chirp.mat* as the noise source n(k). These audio signals were sampled at 8200 Hz. The non-linear passage dynamics was a third order equation. To model f(.) using ANFIS, we collected 1000 training points of the following form:

$$[n(k), n(k\text{-}1), n(k\text{-}2); y(k)], \qquad (17.7)$$

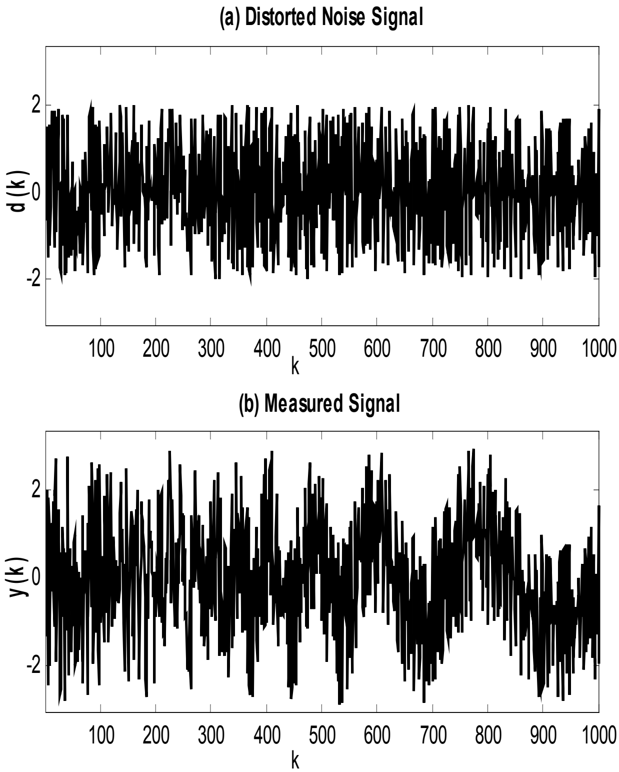


Fig. 17.6. (a) Distorted noise signal d(k), (b) measurable output signal y(k)

with k runs from 2 to 1001. We used an eight rule ANFIS to fit the training data, in which each of the three inputs was assigned two generalized bell membership functions.

We describe now some simulation results with the ANFIS approach for adaptive noise cancellation. In our experiment, we have an information signal x(k) shown in Figure 17.5(a) and noise signal shown in Figure 17.5(b). The measurable noise source is Gaussian with zero mean and unity variance. The resulting distorted noise d(k) produced by the non-linear dynamics is shown in Figure 17.6(a). The measurable signal at the receiving end, denoted by y(k), is equal to the sum of x(k) and d(k), which is shown in Figure 17.6(b).

We can appreciate from Figure 17.6 that ANFIS (for a type-1 fuzzy system) gives good results for noise cancellation. We also have similar results for the second experiment.

17.4 Modified ANFIS Approach for the Type-2 Fuzzy System of Noise Cancellation

If we now want to design an interval type-2 fuzzy system that is capable of adaptive noise cancellation, we need to have an extension of the previous type-1 ANFIS approach. If we assume an interval approximation for the type-2 fuzzy sets, we can use a finite number of type-1 fuzzy systems to construct the type-2 fuzzy system. This means that we can use a finite number of type-1 ANFIS systems for achieving an adaptive type-2 fuzzy system. In the simplest situation, at least two type-1 ANFIS systems are needed to have an interval type-2 ANFIS. This can expressed as follows:

$$\text{Type-2 ANFIS} = \text{Type-1 ANFIS(L)} + \text{Type 1 ANFIS(H)} \quad (17.8)$$

where L stands for low and H for high. Of course, this will only be a good approximation when simple interval type-2 fuzzy sets are used. For Gaussian type-2 fuzzy sets, we need a higher number of type-1 ANFIS systems to have a good approximation.

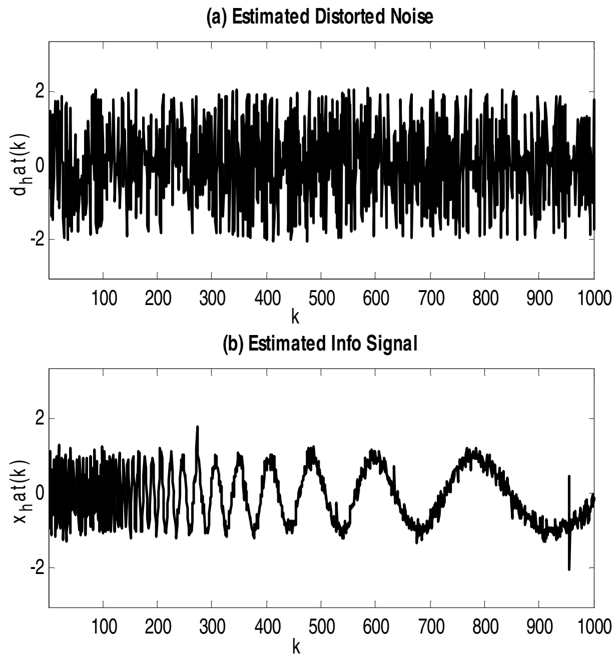


Fig. 17.7. (a) Estimated distorted signal, (b) estimated information signal

For the experiments, we decided to use 4 type-1 ANFIS systems to approximate the Type-2 ANFIS system for adaptive noise cancellation. All the remaining parameters were the same as the ones used in the type-1 case.

Now we show the results of using the type-2 fuzzy system for noise cancellation (trained with ANFIS). We show in Figure 17.7(a) the estimated distorted noise resulting from the non-linear dynamics. We also show in Figure 17.7(b) the estimated information signal. Finally, we show in Figure 17.8(a) the estimated error and in Figure 17.8(b) the original information signal. We can appreciate that the estimated information signal is very close to the original one. For this reason, the fuzzy logic approach is successful for this application.

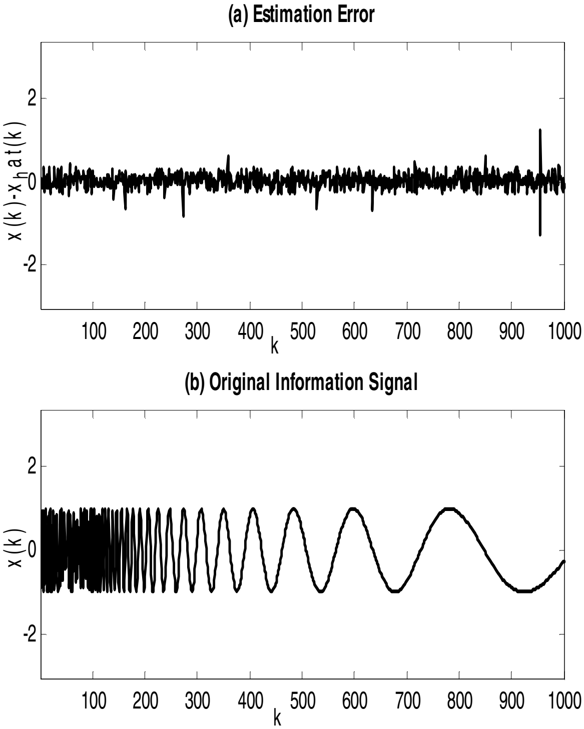


Fig.17. 8. (a) estimated error, (b) original information signal x(k)

17.5 Comparison of Results

We describe in this section the comparison of the simulation results of both approaches. The two experiments described before, were considered by both the type-1 fuzzy logic approach and the type-2 fuzzy logic approach. As a performance measure between the noise cancellation approaches we will use the sum of squared errors between the real information signal and the estimated signal. This can be expressed as follows

$$SSE = \Sigma\ ([e(k)]^2) = \Sigma[y(k) - \mathbf{d(k)}]^2 \quad (17.9)$$

where the sum is over the whole range of points used in the corresponding case. The SSE value gives an idea of how well we are fitting the f function of the non-linear dynamics. We show in Table 17.1 the comparison (based on the average SSE values) of the type-1 and the type-2 fuzzy logic approaches for the two experiments, considered in this paper. We considered 30 runs of ANFIS in each case to obtain an average representative value.

Table 17.1. Comparison between the type-1 and type-2 fuzzy logic approaches

	Type-1	Type-2
Experiment 1	0.1556	0.1445
Experiment 2	0.7732	0.6395

For the first experiment, we have very good noise cancellation results, which is expected due to the simpler nature of the non-linear dynamics added to the noise. Also, we can also say that the difference bettween SSE values is not so great due to the same reason.

From Table 17.1, we can appreciate that the type-2 fuzzy approach outperforms the type-1 approach for noise cancellation in both experiments (based on the SSE values).
For the first experiment, we have very good noise cancellation results, which is expected due to the simpler nature of the non-linear dynamics added to the noise. Also, we can also say that the difference bettween SSE values is not so great due to the same reason.

From Table 17.1, we can say that for the second experiment it is more difficult to perform noise cancellation. Even with the type-2 fuzzy logic approach, we have some dynamics that were not identified correctly, and for this reason the estimated information signal is not so close to the original information signal. Of course, in this case the noise is not Gaussian, as in the previous case. The "chirp" noise is more like an animal sound, which is added to the music of Handel. We believe that the parameters of the type-2 fuzzy system need to be optimized even more. Other methods, like genetic algorithms could be used in the future to try in perform a better optimization of the parameters in the fuzzy system.

Of course, the above comparison is not considering the time of computation involved in the training, which is always higher for the type-2 approach because we are using a set of type-1 ANFIS systems to approximate the type-2 fuzzy sytem. In our particular case, we have used four type-1 ANFIS systems to have an approximation of the type-2 fuzzy system. For this reason, the computation time is approximately four times for the type-2 fuzzy system.

17.6 Summary

We have described in this chapter the application of type-2 fuzzy logic and neural networks (by using ANFIS) for achieving adaptive noise cancellation. Type-2 fuzzy logic was needed because of the high level of uncertainty in modeling the non-linear dynamics that is changing the noise dynamics. Neural networks are needed because we want to use the time series of the signals to adapt the parameters of the fuzzy system. The type-2 fuzzy logic approach was compared against the type-1 approach in two experiments. The simulation results in the experiments, with the proposed approach, show the potential of type-2 fuzzy logic techniques in the field of adaptive noise cancellation.

Bibliography

Akamatsu, S., Saski, T., Fukamachi, H., Masui, N., Suenaga, Y.: Robust Face Identification Scheme. In: Proceedings of SPIE, pp. 71–84 (1992)

Alter, T.D.: 3d pose from 3 corresponding points under weak-perspective projection, Technical Report 1378, MIT Artificial Intelligence Laboratory (1992)

Astudillo, L., Castillo, O., Aguilar, L.T.: Intelligent Control of an Autonomous Mobile Robot Using Type-2 Fuzzy Logic. In: ICAI 2006. Proc. Conference on Artificial Intelligence, Las Vegas, USA, pp. 565–570 (2006)

Atal, B.S.: Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. Journal of the Acoustical Society of America 55(6), 1304–1312 (1974)

Baker, J.E.: Reducing Bias and Inefficiency in the Selection Algorithms. In: Proceedings of the 2nd International Conference Genetic Algorithms, Hillsdale, NJ, pp. 14–21 (1987)

Barna, G., Kaski, K.: Choosing optimal network structure. In: INNC 1990. Proceedings of the International Neural Network Conference, pp. 890–893 (1990)

Barr, J.M.: Speeded Phase Discrimination: Evidence for Global to Local Processing. In: Proceedings of Graphics Interface, pp. 331–336 (1986)

Barto, A.G., Sutton, R.S., Anderson, C.: Neuron like Elements that can Solve Difficult Learning Control Problems. IEEE Transactions on Systems, Man & Cybernetics 13, 835–846 (1983)

Bentalba, S., El Hajjaji, A., Rachid, A.: Fuzzy Control of a Mobile Robot: A New Approach. In: Proc. IEEE Int. Conf. On Control Applications, Hartford, CT, pp. 69–72 (October 1997)

Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)

Bloch, A.M., Drakunov, S.: Tracking in NonHolonomic Dynamic System Via Sliding Modes. In: Proc. IEEE Conf. On Decision & Control, Brighton, UK, pp. 1127–1132 (1991)

Boers, E., Kuiper, H.: Biological Metaphors and the Design of Modular Neural Networks, Thesis, Department of Computer Science, Leiden University, The Netherlands (1992)

Campion, G., Bastin, G., D'Andrea-Novel, B.: Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. IEEE Trans. On Robotics and Automation 12(1) (February 1996)

Cantú-Paz, E.: A Summary of Research on Parallel Genetic Algorithms, IlliGAL Report No. 95007, Illinois Genetic Algorithm Laboratory, University of Illinois at Urbana-Champaing (1995)

Carpenter, G.A., Grossberg, S.: The Art of Adaptive Pattern Recognition by a Self-Organizing Neural Network. IEEE Computer 21(3), 77–88 (1988)

Castillo, O., Melin, P.: Developing a New Method for the Identification of Microorganisms for the Food Industry using the Fractal Dimension. Journal of Fractals 2(3), 457–460 (1994)

Castillo, O., Melin, P.: Automated Mathematical Modelling for Financial Time Series Prediction using Fuzzy Logic, Dynamical System Theory and Fractal Theory. In: Proceedings of CIFE 1996, pp. 120–126. IEEE Press, New York (1996)

Castillo, O., Melin, P.: A new Fuzzy-Fractal-Genetic Method for Automated Mathematical Modelling and Simulation of Robotic Dynamic Systems. In: Proceedings of FUZZ 1998, vol. 2, pp. 1182–1187. IEEE press, Alaska, USA (1998)

Castillo, O., Melin, P.: Automated Mathematical Modelling for Financial Time Series Prediction Combining Fuzzy Logic and Fractal Theory. In: Kacprzyk, J. (ed.) Soft Computing for Financial Engineering, pp. 93–106. Springer, Heidelberg (1999)

Castillo, O., Melin, P.: Soft Computing for Control of Non-Linear Dynamical Systems. Springer, Heidelberg (2001)

Castillo, O., Melin, P.: Hybrid Intelligent Systems for Time Series Prediction using Neural Networks. Fuzzy Logic and Fractal Theory IEEE Transactions on Neural Networks 13(6), 1395–1408 (2002)

Castillo, O., Melin, P.: A Hybrid Fuzzy-Fractal Approach for Time Series analysis and Plant Monitoring. International Journal of Intelligent Systems 17(8), 751–765 (2002)

Castillo, O., Melin, P.: A New Approach for Plant Monitoring using Type-2 Fuzzy Logic and Fractal Theory. In: Proceedings of FLINS 2002, pp. 304–313. World Scientific, Gent, Belgium (2002)

Castillo, O., Melin, P.: Soft Computing and Fractal Theory for Intelligent Manufacturing. Springer, Heidelberg (2003)

Castillo, O., Melin, P.: A New Approach for Plant Monitoring using Type-2 Fuzzy Logic and Fractal Theory. International Journal of General Systems 33, 305–319 (2004)

Castillo, O., Melin, P., Kacprzyk, J., Pedrycz, W.: Hybrid Intelligent Systems: Design and Analysis. Springer, Germany (2007)

Castillo, O., Huesca, G., Valdez, F.: Evolutionary Computing for Fuzzy System Optimization in Intelligent Control. In: Proceedings of ICAI 2004, Las Vegas, USA, pp. 98–104 (2004)

Castillo, O., Aguilar, L., Cazarez, N., Rico, D.: Intelligent Control of Dynamical Systems with Type-2 Fuzzy Logic and stability Study. In: ICAI 2005. Proceedings of the Conference on Artificial Intelligence, pp. 400–405 (2005)

Castro, J.R.: Hybrid Intelligent Architecture Development for Time Series Forecasting. Masters Degree Thesis. Tijuana Institute of Technology (December 2005)

Castro, J.R., Castillo, O., Martínez-Méndez, L.G.: Tool Box para Lógica Difusa Tipo-2 por Intervalos. In: International Seminar on Computational Intelligence 2006 IEEE - CIS Mexico Chapter, October 9-11, 2006, Tijuana, Mexico (2006) (CD-rom proceedings)

Cázarez, N.R., Cárdenas, S., Aguilar, L., Castillo, O.: Lyapunov Stability on Type-2 Fuzzy Logic Control. In: IEEE-CIS International Seminar on Computational Intelligence, October 17-18, 2005, México Distrito Federal, México (2005) (Proceedings in CD-rom)

Chalmers, D.: The evolution of learning: an experiment in genetic connectionism. In: Proceedings of the 1990 Connectionist Models Summer School, Morgan Kaufmann, San Francisco (1990)

Chellapa, R.: Human and Machine Recognition of Faces: A Survey, Technical Report CAR-TR-731, University of Maryland Computer Vision Laboratory (1994)

Chen, G., Pham, T.T.: Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems. CRC Press, Boca Raton, Florida, USA (2001)

Chen, V.C., Pao, Y.H.: Learning Control with Neural Networks. In: Proceedings of the International Conference on Robotics and Automation, pp. 1448–1453 (1989)
Chiang, L.H., Russell, E.L., Braatz, R.D.: Fault Detection and Diagnosis in Industrial Systems. Springer, Heidelberg (2000)
Cho, S.-B.: Fusion of neural networks with fuzzy logic and genetic algorithms. Journal of Integrated Computer Aided Engineering 9, 363–372 (2002)
Choi, B.J., Kwak, S.W., Kim, B.K.: Design and Stability Analysis of Single-Input Fuzzy Logic Controller. IEEE Trans. Fuzzy Systems 30, 303–309 (2000)
Chuang, M.-M., Chang, R.-F., Huang, Y.-L.: Automatic, Facial Feature Extraction in Model-based Coding. Journal of Information Science and Engineering 16, 447–458 (2000)
Chwa, D.: Sliding-Mode Tracking Control of Nonholonomic Wheeled Mobile Robots in Polar coordinates. IEEE Trans. On Control Syst. Tech. 12(4), 633–644 (2004)
Colombo, C., Del Bimbo, A., De Magistris, S.: Human-Computer Interaction Based on Eye Movement Tracking. Computer Architectures for Machine Perception, pp. 258–263 (1995)
Cox, I.J., Ghosn, J., Yianilos, P.N.: Feature-Based Face Recognition using Mixture-Distance, Technical Report TR-95-09, NEC Research Institute (1995)
Craw, I., Tock, D., Bennett, A.: Finding Face Features. In: Proceedings of the Second European Conference on Computer Vision, pp. 92–96 (1992)
Denker, J.S.: Neural Networks for Computation. In: Denker, J.S. (ed.) AIP Conference Proceedings, p. 151. American Institute of Physics (1986)
Deshpande, P.B., Ash, R.: Computer Process Control with Advanced Control Applications, 2nd edn. Instrument Society of America, North Carolina (1988)
Diao, Y., Passino, M.: Adaptive Neural/Fuzzy Control for Interpolated Nonlinear Systems. IEEE Trans. on Fuzzy Systems 10(5), 583–595 (2002)
Dickason, V.: The Loudspeaker Cookbook. McGraw Hill, New York (1997)
Doctor, F., Hagras, H., Callaghan, V.: A type-2 fuzzy embedded agent to realize ambient intelligence in ubiquitous computing environments. Information Sciences 171, 309–334 (2005)
Du, R.: Engineering Monitoring and Diagnosis. In: Leondes, C.T. (ed.) Industrial and Manufacturing Systems, pp. 119–144. Academic Press, San Diego, California, USA (1998)
Du, R., Elbestawi, M.A., Wu, S.M.: Automated Monitoring of Manufacturing Processes. Transactions ASME J. Eng. Ind., 245–252 (1993)
Fahlman, S.E.: Faster-learning Variations on Back-propagation: an empirical study. In: Proceedings of the Conectionist Models Summer School, pp. 38–51. Carnegic Mellon University (1988)
Fahlman, S.E., Lebiere, C.: The Cascade-Correlation Learning Architecture. In: Advances in Neural Information Processing Systems. Morgan Kaufmann, San Francisco (1990)
Fantoni, I., Lozano, R.: Non-Linear Control for Underactuated Mechanical Systems. Springer, London (2002)
Feldman, J.: Neural representation of conceptual knowledge. In: Nadel, et al. (eds.) Neural connections, mental computation, MIT Press, Cambridge (1989)
Fierro, R., Lewis, F.L.: Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics. In: Proc. 34th Conf. on Decision Control, New Orleans, USA (1995)
Fierro, R., Lewis, F.L.: Control of a Nonholonomic Mobile Robot Using Neural Networks. IEEE Trans. On Neural Networks 9(4), 589–600 (1998)
Fogelman-Soulie, F.: Multi-modular neural network-hybrid architectures: a review. In: Proceedings of 1993 International Joint Conference on Neural Networks (1993)
Fukao, T., Nakagawa, H., Adachi, N.: Adaptive Tracking Control of a NonHolonomic Mobile Robot. IEEE Trans. On Robotics and Automation 16(5), 609–615 (2000)

Fu, H.-C., Lee, Y.-P., Chiang, C.-C., Pao, H.-T.: Divide-and-Conquer Learning and Modular Perceptron Networks. IEEE Transactions on Neural Networks 12(2), 250–263 (2001)

Furui, S.: Cepstral analysis technique for automatic speaker verification. IEEE Transactions on Acoustics, Speech and Signal Processing 29(2), 254–272 (1981)

Furui, S.: Research on individuality features in speech waves and automatic speaker recognition techniques. Speech Communication 5(2), 183–197 (1986)

Furui, S.: Speaker-independent isolated word recognition using dynamic features of the speech spectrum. IEEE Transactions on Acoustics, Speech and Signal Processing 29(1), 59–59 (1986)

Furui, S.: Digital Speech Processing, Synthesis, and Recognition. Marcel Dekker, New York (1989)

Furui, S.: Speaker-dependent-feature extraction, recognition and processing techniques. Speech Communication 10(5-6), 505–520 (1991)

Furui, S.: An overview of speaker recognition technology. In: Proceedings of the ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, pp. 1–9 (1994)

Gillies, A.M.: Machine Learning Procedures for Generating Image Domain Feature Detectors, PhD thesis, University of Michigan (1985)

Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading (1989)

Graf, H.P., Chen, T., Petajan, E., Cosatto, E.: Locating Faces and Facial Parts. In: Proceedings of the International Workshop on Automatic Face and Gesture Recognition, pp. 41–46 (1995)

Hagras, H.A.: Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots. IEEE Transactions on Fuzzy Systems 12(4), 524–539 (2004)

Haykin, S.: Adaptive Filter Theory, 3rd edn. Prentice Hall, Englewood Cliffs (1996)

Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(10), 993–1001 (1990)

Happel, B., Murre, J.: Design and evolution of modular neural network architectures. Neural Networks 7, 985–1004 (1994)

Higgins, A.L., Bahler, L., Porter, J.: Speaker verification using randomized phrase prompting. Digital Signal Processing 1, 89–106 (1991)

Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)

Homaifar, A., McCormick, E.: Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers using Genetic Algorithms. IEEE Transactions on Fuzzy Systems 3, 129–139 (1995)

Hopfield, J.J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. In: Proceedings of the National Academy of Science, USA, pp. 2554–2558 (1982)

Hollstien, R.B.: Artificial Genetic Adaptation in Computer Control Systems, PhD thesis, University of Michigan (1971)

Hsu, C.: Advanced Signal Processing Technology by Soft Computing. World Scientific Press, Singapore (2000)

Hunt, K.J., Sbarbaro, D., Zbikowski, R., Gawthrop, P.J.: Neural Networks for Control Systems-A survey. Automatica 28(6), 1083–1112 (1992)

Ingle, V.K., Proakis, J.G.: Digital Signal Processing using MATLAB. Brooks/Cole Publishing Company, Pacific Grove, CA (2000)

Ishikawa, S.: A Method of Indoor Mobile Robot Navigation by Fuzzy Control. In: Proc. Int. Conf. Intell. Robot. Syst., Osaka, Japan, pp. 1013–1018 (1991)

Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive Mixtures of Local Experts. Neural Computation 3, 79–87 (1991)
Jamshidi, M.: Large-Scale Systems: Modelling, Control and Fuzzy Logic. Prentice-Hall, Englewood Cliffs (1997)
Jain, A., Jain, A., Jain, S., Jain, L.: Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis. World Scientific Press, Singapore (2000)
Jang, J.-S.R.: ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. IEEE Transactions on Systems, Man and Cybernetics 23, 665–685 (1993)
Jang, J.-S.R., Gulley, N.: MATLAB: Fuzzy Logic Toolbox, User's Guide. The Math-Works, Inc. Publisher (1997)
Jang, J.-S.R., Sun, C.-T., Mizutani, E.: Neurofuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Englewood Cliffs (1997)
Janikow, C.Z., Michalewicz, Z.: An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms. In: Proceedings of 4th International Conference Genetic Algorithms, pp. 31–36 (1991)
Jenkins, R., Yuhas, B.: A simplified neural network solution through problem decomposition: The case of the truck backer-upper. IEEE Transactions on Neural Networks 4(4), 718–722 (1993)
Jordan, M., Jacobs, R.: Hierarchical Mixtures of Experts and the EM Algorithm. Journal of Neural Computations 5, 181–214 (1994)
Kanade, T.: Picture Processing System by Computer Complex and Recognition of Human Faces, Ph.D. Thesis, Kyoto University, Japan (1973)
Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T.: A Stable Tracking Control Method For a Non-Holonomic Mobile Robot. In: Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, Osaka, Japan, pp. 1236–1241 (1991)
Kandel, A.: Fuzzy Expert Systems. CRC Press Inc., Boca Raton, USA (1992)
Karnik, N.N., Mendel, J.M.: An Introduction to Type-2 Fuzzy Logic Systems, Technical Report, University of Southern California (1998)
Karnik, N.N., Mendel, J.M.: Applications of Type-2 Fuzzy Logic Systems to Forecasting of Time-series. Information Sciences 120(1-4), 89–111 (1999)
Karnik, N.N., Mendel, J.M.: Operations on type-2 fuzzy sets, Signal and Image Processing Institute, Department of Electrical Engineering-Systems, University of Southern California (2000)
Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. Information Sciences 132(1-4), 195–220 (2001)
Karnik, N.N., Mendel, J.M.: Operations on type-2 fuzzy sets. International Journal on Fuzzy Sets and Systems 122(2), 327–348 (2001)
Karnik, N.N., Mendel, J.M., Liang, Q.: Type-2 Fuzzy Logic Systems. IEEE Transactions on Fuzzy Systems 7(6), 643–658 (1999)
Karr, C.L., Gentry, E.J.: Fuzzy Control of pH using Genetic Algorithms. IEEE Transactions on Fuzzy Systems 1, 46–53 (1993)
Kelly, M.F.: Annular Symmetry Operators: A Multi-Scalar for Locating and Describing Imaged Objects, Ph.D. Thesis, McGill University, Canada (1995)
Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)
Kitano, H.: Designing neural networks using genetic algorithms with graph generation system. Journal of Complex Systems 4, 461–476 (1990)
Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic Theory and Applications. Prentice Hall, New York (1995)

Kolmanovsky, I., McClamroch, N.H.: Developments in Nonholonomic Nontrol Problems. IEEE Control Syst. Mag. 15, 20–36 (1995)

Kosko, B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, Englewood Cliffs (1992)

Kosko, B.: Fuzzy Engineering. Prentice-Hall, Englewood Cliffs (1997)

Kuncheva, L.I.: Fuzzy Versus Nonfuzzy in Combining Classifiers Designed by Boosting. IEEE Transactions on Fuzzy Systems 11(6), 729–741 (2003)

Lee, C.-H., Hong, J.-L., Lin, Y.-C., Lai, W.-Y.: Type-2 Fuzzy Neural Network Systems and Learning. International Journal of Computational Cognition 1(4), 79–90 (2003)

Lee, S., Kil, R.M.: A Gaussian Potential Function Network with Hierarchically Self-Organizing Learning. Neural Networks 4(2), 207–224 (1991)

Lee, J.: On methods for improving performance of PI-type fuzzy logic controllers. IEEE Trans. Fuzzy Systems 1, 298–301 (1993)

Lee, T.-C., Lee, C.H., Teng, C.-C.: Tracking Control of Mobile Robots Using the Backsteeping Technique. In: Proc. 5th. Int. Conf. Contr., Automat., Robot. Vision, Singapore, pp. 1715–1719 (1998)

Lee, T.-C., Tai, K.: Tracking Control of Unicycle-Modeled Mobile robots Using a Saturation Feedback Controller. IEEE Trans. On Control Systems Technology 9(2), 305–318 (2001)

Lee, T.H., Leung, F.H.F., Tam, P.K.S.: Position Control for Wheeled Mobile Robot Using a Fuzzy Controller, pp. 525–528. IEEE, Los Alamitos (1999)

Li, H.-X., Gatland, H.B.: A new methodology for designing a fuzzy logic controller. IEEE Trans. Systems Man and Cybernetics 25, 505–512 (1995)

Li, H.-X., Gatland, H.B.: Conventional fuzzy control and its enhancement. IEEE Trans. Systems Man and Cybernetics 25, 791–797 (1996)

Liang, Q., Mendel, J.M.: Interval Type-2 Fuzzy Logic Systems: Theory and Design. IEEE Transactions on Fuzzy Systems 8(5), 535–550 (2000)

Lippmann, R.P.: An Introduction to Computing with Neural Networks. IEEE Acoustics, Speech, and Signal Processing Magazine 4, 4–22 (1987)

Co, L.: Speaker Assembly Adhesives Guide (1999)

Lowe, D.: Adaptive Radial Basis Function Nonlinearities, and the Problem of Generalization. In: Proceedings of the First IEEE International Conference on Artificial Neural Networks, London, pp. 171–175 (1989)

Lowe, D.G.: Perceptual Organization and Visual Recognition. Kluwer Academic Publishers, Dordrecht (1985)

Lozano, A.: Optimizacion de un Sistema de Control Difuso por medio de Algoritmos Geneticos Jerarquicos, Thesis, Dept. of Computer Science, Tijuana Institute of Technology, Mexico (2004)

Lu, B.-L., Ito, M.: Task Decomposition and Module Combination Based on Class Relations: Modular Neural Network for Pattern Classification, Technical Report, Nagoya, Japan (1998)

Lu, B.-L., Ito, M.: Task Decomposition and Module Combination Based on Class Relations: A Modular Neural Network for Pattern Classification. IEEE Transactions on Neural Networks 10(5), 1244–1256 (1999)

Madisetti, V.K., Williams, D.B.: The Digital Signal Processing Handbook, CRC Press in cooperation with IEEE Press, Boca Raton, Florida (1998)

Mamdani, E.H.: Application of fuzzy algorithms for control of a simple dynamic plant. Proceedings of the IEEE 121, 1585–1588 (1974)

Mamdani, E.H., Assilian, S.: An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. International Journal of Man-Machine Studies 7, 1–13 (1975)

Mamdani, E.H.: Twenty years of Fuzzy Control: Experiences Gained and Lessons Learned. In: Proc. 2nd IEEE International Conference on Fuzzy Systems, San Francisco, CA, pp. 339–344 (1993)

Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms. Springer, Heidelberg (1999)

Mandelbrot, B.: The Fractal Geometry of Nature. W.H. Freeman and Company, New York (1987)

Mandelbrot, B.: Fractals and Scaling in Finance. Springer, Heidelberg (1997)

Manjunath, B.S., Chellappa, R., Malsburg, C.V.D.: A Feature Based Approach to Face Recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision, pp. 373–378 (1992)

Margaliot, M., Langholz, G.: New Approaches to Fuzzy Modeling and Control. World Scientific Press, Singapore (2000)

Masters, T.: Practical Neural Network recipe in C++. Academic Press, Inc., London (1993)

Matsui, T., Furui, S.: Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMMs. In: Proceedings of ICASSP 1993, pp. 157–160 (1993)

Matsui, T., Furui, S.: Concatenated phoneme models for text-variable speaker recognition. In: Proceedings of ICASSP 1993, pp. 391–394 (1993)

Matsui, T., Furui, S.: Similarity normalization method for speaker verification based on a posteriori probability. In: Proceedings of the ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, pp. 59–62 (1994)

Matsui, T., Furui, S.: Speaker adaptation of tied-mixture-based phoneme models for text-prompted speaker recognition. In: Proceedings of ICASSP 1994, pp. 125–128 (1994)

Melin, P., Castillo, O.: An Adaptive Model-Based Neuro-Fuzzy-Fractal Controller for Biochemical Reactors in the Food Industry. In: Proceedings of IJCNN 1998, pp. 106–111. IEEE Computer Society Press, Alaska, USA (1998)

Melin, P., Castillo, O.: A New Method for Adaptive Model-Based Control of Non-Linear Dynamic Plants using a Neuro-Fuzzy-Fractal Approach. Soft Computing 5(2), 171–177 (2001)

Melin, P., Castillo, O.: Modelling Simulation and Control of Non-Linear Dynamical Systems: An Intelligent Approach Using Soft Computing and Fractal Theory. Taylor and Francis, London (2002)

Melin, P., Acosta, M.L., Felix, C.: Pattern Recognition Using Fuzzy Logic and Neural Networks. In: Proceedings of ICAI 2003, Las Vegas, USA, pp. 221–227 (2003)

Melin, P., Castillo, O.: A new method for adaptive model-based control of non-linear dynamic plants using type-2 fuzzy logic and neural networks. In: Proc. 12th IEEE International Conference on Fuzzy Systems, pp. 420–425. St. Louis, Missouri (2003)

Melin, P., Castillo, O.: A New Method for Adaptive Control of Non-Linear Plants Using Type-2 Fuzzy Logic and Neural Networks. International Journal of General Systems 33, 289–304 (2004)

Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing. Springer, Heidelberg (2005)

Melin, P., Gonzalez, F., Martinez, G.: Pattern Recognition Using Modular Neural Networks and Genetic Algorithms. In: Proceedings of ICAI 2004, Las Vegas, USA, pp. 77–83 (2004)

Melin, P., Mancilla, A., Gonzalez, C., Bravo, D.: Modular Neural Networks with Fuzzy Sugeno Integral Response for Face and Fingerprint Recognition. In: Proceedings of ICAI 2004, Las Vegas, USA, pp. 91–97 (2004)

Melin, P., Mancilla, A., Lopez, M., Mendoza, O.: A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting. Journal of Applied Soft Computing (accepted, 2007)

Mendel, J.M.: Type-2 fuzzy logic systems: Type-reduction. In: IEEE Syst., Man, Cybern. Conf., San Diego, CA, pp. 2046–2051 (1998)
Mendel, J.M.: Computing with words, when words can mean different things to different people. In: Int. ICSC Congress Computat. Intell. Methods Applications, Rochester, NY (June 1999)
Mendel, J.M.: Uncertainty, fuzzy logic, and signal processing. Signal Processing Journal 80, 913–933 (2000)
Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, New Jersey, USA (2001)
Mendel, J.M.: Computing Derivatives in Interval Type-2 Fuzzy Logic Systems. IEEE Transactions on Fuzzy Systems 12, 84–98 (2004)
Mendel, J.M.: Computing Derivatives in Interval Type-2 Fuzzy Logic Systems. IEEE Transactions on Fuzzy Systems 12, 84–98 (2004)
Mendel, J.M.: On a 50 % savings in the computation of the centroid of a symmetrical interval type-2 fuzzy set. Information Sciences 172(3-4), 417–430 (2005)
Mendel, J.M., Mouzouris, G.C.: Type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems 7, 643–658 (1999)
Mendel, J.M., John, R.I.B.: Type-2 Fuzzy Sets Made Simple. IEEE Transactions on Fuzzy Systems 10(2), 117–127 (2002)
Mendel, J.M., Wu, H.: Properties of the Centroid of an Interval Type-2 Fuzzy Set, Including the Centroid of a Fuzzy Granule. In: Proc. FUZZ-IEEE 2005, Reno, NV, pp. 341–346 (2005)
Mendoza, O., Melin, P.: The Fuzzy Sugeno Integral as a Decision Operator in the Recognition of Images with Modular Neural Networks. In: International Conference on Fuzzy Systems, Neural Networks and Genetic Algorithms FNG 2005 (2005) (Proceedings in CD-rom)
Mendoza, O., Melin, P.: Sistemas de Inferencia Difusos Tipo-1 y Tipo-2 Aplicados a la Detección de Bordes en Imágenes Digitales. In: International Seminar on Computational Intelligence 2006 IEEE - CIS Mexico Chapter, Tijuana Institute of Technology, Tijuana, Mexico (2006) (CD-rom proceedings)
Mendoza, O., Melin, P.: The Fuzzy Sugeno Integral as a Decision Operator in the Recognition of Images with Modular Neural Networks. In: Hybrid Intelligent Systems: Design and Analysis, pp. 299–310. Springer, Heidelberg (2007)
Messner, W.C., Tilbury, D.M.: Control Tutorials for MatLab and Simulink, a web Based Approach. Prentice Hall, New Jersey (1999)
Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Program, 3rd edn. Springer, Heidelberg (1996)
Miller, W.T., Sutton, R.S., Werbos, P.J.: Neural Networks for Control. MIT Press, Boston, MA, USA (1995)
Miller, G., Todd, P., Hedge, S.: Designing Neural Networks using Genetic Algorithms. In: Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kauffmann, San Francisco (1989)
Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1996)
Mizumoto, M., Tanaka, K.: Some properties of fuzzy sets of type-2. Information and Control 31, 312–340 (1976)
Moghaddam, B., Nastar, C., Pentland, A.: Bayesian Face Recognition Using Deformable Intensity Surfaces, Technical Report 371, MIT Media Laboratory, Perceptual Computing Section (1996)
Monrocq, C.: A probabilistic approach which provides and adaptive neural network architecture for discrimination. In: Proceedings of the International Conference on Artificial Neural Networks, vol. 372, pp. 252–256 (1993)

Montana, G., Davis, L.: Training feedforward networks using genetic algorithms. In: Proceedings of the International Joint Conference on Artificial Intelligence, Morgan Kauffmann, San Francisco (1989)

Montiel, O., Castillo, O., Melin, P., Rodríguez, A., Sepúlveda, R.: The Human Evolutionary Model. Journal of Intelligent Systems. Special Issue: Hybrid Intelligent Systems For Time Series Prediction 14(2-3), 213–236 (2005)

Moody, J., Darken, C.: Fast Learning in Networks of Locally-Tuned Processing Units. Neural Computation 1, 281–294 (1989)

Moon, F.C.: Chaotic and Fractal Dynamics: An Introduction for Applied Scientists and Engineers. Wiley-Interscience, New York (1992)

Moulet, M.: A Symbolic Algorithm for Computing Coefficients Accuracy in Regression. In: Proceedings of the International Workshop on Machine Learning, pp. 332–337 (1992)

Mühlenbein, H., Schilierkamp, V.: Predictive Model for Breeder Genetic Algorithm. Evolutionary Computation 1(1), 25–49 (1993)

Murray-Smith, R., Johansen, T.A.: Multiple Model Approaches for Modeling and Control. Taylor and Francis, United Kingdom (1997)

Naik, J.M., Netsch, L.P., Doddington, G.R.: Speaker verification over long distance telephone lines. In: Proceedings of ICASSP 1989, pp. 524–527 (1989)

Nakamura, S.: Numerical Analysis and Graphic Visualization with MATLAB. Prentice Hall, Englewood Cliffs (1997)

Nastar, C., Pentland, A.: Matching and Recognition Using Deformable Intensity Surfaces. In: Proceedings of the International Symposium on Computer Vision, pp. 223–228 (1995)

Nelson, W., Cox, I.: Local Path Control for an Autonomous Vehicle. In: Proc. IEEE Conf. On Robotics and Automation, pp. 1504–1510 (1988)

Nixon, M.: Eye Spacing Measurement for Facial Recognition. In: SPIE Proceedings, pp. 279–285 (1985)

Ogata, K.: Ingeniería de Control Moderna, 3rd edn. Prentice Hall, Hispanoamericana, Spain (1998)

Omidvar, O., Elliot, D.L.: Neural Systems for Control. Academic Press, San Diego, CA, USA (1997)

Ozen, T., Garibaldi, J.M.: Investigating Adaptation in Type-2 Fuzzy Logic Systems Applied to Umbilical Acid-Base Assessment. In: EUNITE 2003. Proc. European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, Oulu, Finland (2003)

Parker, D.B.: Learning Logic, Invention Report 581-64, Stanford University (1982)

Passino, K., Yurkovich, S.: Fuzzy Control. Addison-Wesley, New York (1998)

Patton, R.J., Frank, P.M., Clark, R.N.: Issues of Fault Diagnosis for Dynamic Systems. Springer, Heidelberg (2000)

Paul, H., Yang, C.: Sistemas de control en ingeniería. Prentice Hall, Iberia, Spain (1999)

Pawlowski, S., Dutkiewicz, P., Kozlowski, K., Wroblewski, W.: Fuzzy Logic Implementation in Mobile Robot Control. In: 2nd Workshop On Robot Motion and Control, pp. 65–70 (2001)

Peitgen, H.-O., Saupe, D., Jurgens, H., Yunker, L.: Chaos and Fractals: New Frontiers of Science. Springer, Heidelberg (1992)

Pentland, A., Moghaddam, B., Starner, T.: View-Based and Modular Eigenspaces for Face Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 84–91 (1994)

Pesin, Y.B.: Dimension Theory in Dynamical Systems: Contemporary Views and Applications. University of Chicago Press, USA (1999)

Peters, E.: Fractal Market Analysis: Applying Chaos Theory to Investment and Economics. John Wiley & Sons, New York (1994)

Pham, D.T., Xing, L.: Neural Networks for Identification, Prediction and Control. Springer, Heidelberg (1995)

Pomerleau, D.A.: Efficient Training of Artificial Neural Networks for Autonomous Navigation. Journal of Neural Computation 3, 88–97 (1991)

Powell, M.J.D.: Radial Basis Functions for Multivariable Interpolation: a review. In: Mason, J.C., Cox, M.G. (eds.) Algorithms for approximation, pp. 143–167. Oxford University Press, Oxford (1987)

Proakis, J.G., Manolakis, D.G.: Digital Signal Processing Principles, Algorithms, and Applications, 3rd edn. Prentice Hall, New Jersey (1996)

Procyk, T.J., Mamdani, E.M.: A Linguistic Self-Organizing Process Controller. Automatica 15(1), 15–30 (1979)

Psaltis, D., Sideris, A., Yamamura, A.: A Multilayered Neural Network Controller. IEEE Control Systems Magazine 8, 17–21 (1988)

O'Shaughnessy, D.: Speaker recognition. IEEE Acoustics, Speech and Signal Processing Magazine 3(4), 4–17 (1986)

Quezada, A.: Reconocimiento de Huellas Digitales Utilizando Redes Neuronales Modulares y Algoritmos Geneticos, Thesis of Computer Science, Tijuana Institute of Technology, Mexico (2004)

Rao, R., Ballard, D.: Natural Basis Functions and Topographics Memory for Face Recognition. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 10–17 (1995)

Rasband, S.N.: Chaotic Dynamics of Non-Linear Systems. John Wiley & Sons, New York (1990)

Reisfeld, D.: Generalized Symmetry Transforms: Attentional Mechanisms and Face Recognition, Ph.D. Thesis, Tel-Aviv University, Israel (1994)

Reisfeld, D., Yeshurun, Y.: Robust Detection of Facial Features by Generalized Symmetry. In: Proceedings of the IAPR International Conference on Pattern Recognition, pp. 117–120 (1992)

Ritter, H., Schulten, K.: Extending Kohonen's Self-Organizing Mapping Algorithm to Learn Ballistic Movements. In: Eckmiller, R., Malsburg, C.v.d. (eds.) Neural Computers, pp. 393–406. Springer, London (1987)

Ronco, E., Gawthrop, P.J.: Modular Neural Networks: State of the Art, Technical Report CSC-95026, Faculty of Engineering, University of Glasgow, United Kingdom (1995)

Rosenberg, A.E., Soong, F.K.: Recent research in automatic speaker recognition. In: Furui, S., Sondhi, M.M. (eds.) Advances in Speech Signal Processing, pp. 701–737. Marcel Dekker, New York (1991)

Rosenblatt, F.: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan (1962)

Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing: Explorations in the Microestructure of Cognition, vol. 1, ch. 8, pp. 318–362. MIT Press, Cambridge (1986)

Rumelhart, D.E., Zipser, D.: Feature Discovery by Competitive Learning. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing: explorations in the microstructure of cognition, vol. 1, ch. 5, pp. 151–193. MIT Press, Cambridge (1986)

Runkler, T.A., Glesner, M.: Defuzzification and Ranking in the Context of Membership Value Semantics, Rule Modality, and Measurement Theory. In: Proceedings of European Congress on Fuzzy and Intelligent Technologies (1994)

Ruspini, E.H.: Recent Development in Fuzzy Clustering, Fuzzy Set and Possibility Theory, pp. 133–147. North Holland, Amsterdam (1982)

Russell, E.L., Chiang, L.H., Braatz, R.D.: Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes. Springer, Heidelberg (2000)

Sackinger, E., Boser, B.E., Bromley, J., LeCun, Y., Jackel, L.D.: Application of the Anna Neural Network Chip to High-Speed Character Recognition. IEEE Transactions on Neural Networks 3, 498–505 (1992)

Saez, D., Cipriano, A.: Optimization of Industrial Processes at Supervisory Level. Springer, Heidelberg (2001)

Scharf, L.L.: Statistical Signal Processing, Detection, Estimation, and Time Series Analysis. Addison-Wesley, Reading (1991)

Schmidt, A., Bandar, Z.: A Modular Neural Network Architecture with Additional Generalization Capabilities for Large Input Vectors. In: Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA), Norwich, England (1997)

Semmes, S.: Some Novel Types of Fractal Geometry. Oxford University Press, England (2000)

Sharkey, A.: Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems. Springer, London (1999)

Song, K.T., Sheen, L.H.: Heuristic fuzzy-neural Network and its application to reactive navigation of a mobile robot. Fuzzy Sets Systems 110(3), 331–340 (2000)

Soucek, B.: Neural and Intelligent Systems Integration: Fifth and Sixth Generation Integrated Reasoning Information Systems. John Wiley and Sons, Chichester (1991)

Spears, W.M., DeJong, K.: An Analysis of Multi-point Crossover. In: Rawlins, G.J.E (ed.) Foundations of Genetic Algorithms, pp. 301–315 (1991)

Staib, W.E.: The Intelligent Arc Furnace: Neural Networks Revolutionize Steelmaking. In: Proceedings of the World Congress on Neural Networks, pp. 466–469 (1993)

Staib, W.E., Staib, R.B.: The Intelligent Arc Furnace Controller: A Neural Network Electrode Position Optimization System for the Electric Arc Furnace. In: Proceedings of the International Conference on Neural Networks, vol. 3, pp. 1–9 (1992)

Su, H.T., McAvoy, T.J.: Neural Model Predictive Models of Nonlinear Chemical Processes. In: Proceedings of the 8th International Symposium on Intelligent Control, pp. 358–363 (1993)

Su, H.T., McAvoy, T.J., Werbos, P.J.: Long-term Predictions of Chemical Processes using Recurrent Neural Networks: A Parallel Training Approach. Industrial & Engineering Chemistry Research 31, 1338–1352 (1992)

Sugeno, M.: Theory of Fuzzy Integrals and its Applications, Ph.D. Thesis, Tokyo Institute of Technology, Japan (1974)

Sugeno, M.: On Stability of Fuzzy Systems Expressed by Fuzzy Rules with Singleton Consequents. IEEE Trans. Fuzzy Systems 7(2) (1999)

Sugeno, M., Kang, G.T.: Structure Identification of Fuzzy Model. Journal of Fuzzy Sets and Systems 28, 15–33 (1988)

Sun, F., Sun, Z., Woo, P.Y.: Neural Network-Based Adaptive Controller Design of Robotic Manipulators with an Observer. IEEE Trans. on Neural Networks 12(1), 54–67 (2001)

Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and its Applications to Modeling and Control. IEEE Transactions on Systems, Man and Cybernetics 15, 116–132 (1985)

Tang, K.-S., Man, K.-F., Liu, Z.-F., Kwong, S.: Minimal Fuzzy Memberships and Rules using Hierarchical Genetic Algorithms. IEEE Transactions on Industrial Electronics 45(1), 162–169 (1998)

Tricot, C.: Curves and Fractal Dimension. Springer, Heidelberg (1995)

Troudet, T.: Towards Practical Design using Neural Computation. In: Proceedings of the International Conference on Neural Networks, vol. 2, pp. 675–681 (1991)

Tsai, C.-C., Lin, H.-H., Lin, C.-C.: Trajectory Tracking Control of a Laser-Guided Wheeled Mobile Robot. In: Proc. IEEE Int. Conf. On Control Applications, Taipei, Taiwan, pp. 1055–1059 (2004)

Tsoukalas, L.H., Uhrig, R.E.: Fuzzy and Neural Approaches in Engineering. John Wiley & Sons, Inc., New Jersey, USA (1997)

Tsukamoto, Y.: An Approach to Fuzzy Reasoning Method. In: Gupta, M.M., Ragade, R.K., Yager, R.R. (eds.) Advanced in Fuzzy Set Theory and Applications, pp. 137–149. North-Holland, Amsterdam (1979)

Turk, M.A., Pentland, A.P.: Face Recognition using Eigenfaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591 (1991)

Ulyanov, S.V., Watanabe, S., Ulyanov, V.S., Yamafuji, K., Litvintseva, L.V., Rizzotto, G.G.: Soft Computing for the Intelligent Robust Control of a Robotic Unicycle with a New Physical Measure for Mechanical Controllability. In: Soft Computing 2, pp. 73–88. Springer, Heidelberg (1998)

Urias, J., Solano, D., Soto, M., Lopez, M., Melin, P.: Type-2 Fuzzy Logic as a Method of Response Integration in Modular Neural Networks. In: Proceedings of the 2006 International Conference on Artificial Intelligence, vol. 2, p. 584 (2006)

Von Altrock, C.: Fuzzy Logic & Neuro Fuzzy Applications Explained. Prentice Hall, Englewood Cliffs (1995)

Wagenknecht, M., Hartmann, K.: Application of Fuzzy Sets of Type 2 to the solution of Fuzzy Equation Systems. Fuzzy Sets and Systems 25, 183–190 (1988)

Wang, L.-X.: A Course in Fuzzy Systems and Control. Prentice-Hall, Upper Saddle River (1997)

Werbos, P.J.: An Overview of Neural Networks for Control. IEEE Control Systems Magazine 11, 40–41 (1991)

Werbos, P.J.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. Thesis, Harvard University (1974)

White, H.: An Additional Hidden Unit Test for Neglected Non-linearity in Multilayer Feedforward Networks. In: Proceedings of IJCNN 1989, pp. 451–455. IEEE Press, Washington, D.C. (1989)

Widrow, B., Glover, J.R.: Adaptive Noise Cancelling: Principles and Applications. IEEE Proceedings 63, 1692–1716 (1975)

Widrow, B., Stearns, D.: Adaptive Signal Processing. Prentice-Hall, Englewood Cliffs (1985)

Wright, A.H.: Genetic Algorithms for Real Parameter Optimization. In: Rawlins, J.E. (ed.) Foundations of Genetic Algorithms, pp. 205–218. Morgan Kaufmann, San Francisco (1991)

Wu, H., Mendel, J.M.: Introduction to Uncertainty bounds and Their Use in the Design of Interval Type-2 Fuzzy Logic Systems. In: Proc. FUZZ-IEEE 2001, Melbourne, Australia, pp. 662–665 (2001)

Yager, R.R.: Fuzzy Subsets of Type II in Decisions. Journal of Cybernetics 10, 137–159 (1980)

Yager, R.R.: A General Approach to Criteria Aggregation Using Fuzzy Measures. International Journal of Man-Machine Studies 1, 187–213 (1993)

Yager, R.R.: Criteria Aggregations Functions Using Fuzzy Measures and the Choquet Integral. International Journal of Fuzzy Systems 1(2) (1999)

Yager, R.R., Filev, D.P.: SLIDE: A Simple Adaptive Defuzzification Method. IEEE Transactions on Fuzzy Systems 1, 69–78 (1993)

Yager, R.R., Filev, D.P.: Generation of Fuzzy Rules by Mountain Clustering. Journal of Intelligent and Fuzzy Systems 2(3), 209–219 (1994)

Yang, C.C., Marefat, M.M., Ciarallo, F.W.: Modeling Errors for Dimensional Inspection using Active Vision. Journal of Robotics and Computer-Integrated Manufacturing 15, 23–27 (1999)

Yen, J., Langari, R., Zadeh, L.A.: Industrial Applications of Fuzzy Control and Intelligent Systems. IEEE Computer Society Press, Los Alamitos (1995)

Yen, J., Langari, R.: Fuzzy Logic: Intelligence, Control, and Information. Prentice Hall, New Jersey, USA (1999)

Ying, H.: Practical Design of Nonlinear Fuzzy Controllers with Stability Analysis for Regulating Processes with Unknown Mathematical Models. Automatica 30(7), 1185–1195 (1994)

Yoshikawa, T., Furuhashi, T., Uchikawa, Y.: Emergence of Effective Fuzzy Rules for Controlling Mobile Robots using DNA Coding Method. In: Proceedings of ICEC 1996, Nagoya, Japan, pp. 581–586 (1996)

Yuille, A., Cohen, D., Hallinan, P.: Feature Extraction from Faces using Deformable Templates. In: Proceedings of IEEE Conference on Computer Vision and Templates, pp. 104–109 (1989)

Zadeh, L.A.: Fuzzy Sets. Journal of Information and Control 8, 338–353 (1965)

Zadeh, L.A.: Similarity Relations and Fuzzy Ordering. Journal of Information Sciences 3, 177–206 (1971a)

Zadeh, L.A.: Quantitative Fuzzy Semantics. Journal of Information Sciences 3, 159–176 (1971b)

Zadeh, L.A.: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Transactions on Systems, Man and Cybernetics 3, 28–44 (1973)

Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning-1. Journal of Information Sciences 8, 199–249 (1975)

Zadeh, L.A.: Fuzzy Sets. Information and Control 8 (1975)

Zadeh, L.A.: Fuzzy Logic. Computer 1(4), 83–93 (1988)

Zadeh, L.A.: Knowledge representation in fuzzy logic. IEEE Transactions on Knowledge and Data Engineering 1, 89–100 (1989)

Zadeh, L.A.: Fuzzy Logic = Computing with Words. IEEE Transactions on Fuzzy Systems 4(2) (1996)

Zadeh, L.A.: Toward a generalized theory of uncertainty (GTU)- an outline. Information Sciences 172, 1–40 (2005)

Zilouchian, A., Jamshidi, M.: Intelligent Control Systems Using Soft Computing Methodologies. CRC Press, Boca Raton, Florida, USA (2001)

Appendix

The Type-2 Fuzzy Logic Toolbox has been developed by our research group in Tijuana, Mexico and can be obtained by sending an email message to Prof. Oscar Castillo (ocastillo@hafsamx.org), and mentioning that it will be used with this book. You will be asked to send a letter in which you formally ask to use the software and after a few months another letter stating that the software has been useful in your research.

This type-2 fuzzy logic toolbox has a similar interface to the type-1 fuzzy logic toolbox already available in Matlab®, and as consequence can be used very easily. You will need to have Matlab® from version 7.1 and up to execute this toolbox. A web site for users of this toolbox will be build in the future so that more help and information will be available to the users.

The goal of the toolbox is to help promote the area of type-2 fuzzy logic, but it is only a tool to help research and academic work, it is not intended for commercial or industrial use and no guarantee can be made of the results obtained with this toolbox. Also, the authors are not responsible of the results or possible damage or injuries to others as a result of using the type-2 fuzzy logic toolbox.

We would like to thank our Ph.D. student (Juan Ramon Castro-Rodriguez) for his great help in developing the computer programs of the type-2 fuzzy logic toolbox. Juan Ramon' thesis work is in the area of type-2 fuzzy logic and for this reason we felt that developing a toolbox for developing type-2 fuzzy systems was very important.

We would like to mention that in most of the chapters of this book, the toolbox was used to apply the concepts of interval type-2 fuzzy logic in solving the problems of pattern recognition, intelligent control and intelligent manufacturing and automation. As a consequence in most of the chapters the use of the toolbox for type-2 fuzzy logic is illustrated. In particular, in Chapter 12 there is a detailed description of the type-2 fuzzy logic toolbox. In this chapter, the use of the toolbox is illustrated with simple examples, so the reader can better understand how to use it.

In conclusion, we recommend using this type-2 fuzzy logic toolbox to learn the theory and concepts of type-2 fuzzy logic. We also recommend using this toolbox for developing intelligent systems that solve problems from a wide rage of areas, like pattern recognition, control, manufacturing, robotics, and automation. It is possible that after using the toolbox for solving a particular problem, it may be necessary that the users develop their own computer programs in the programming language more suitable for the application. In any case, we feel that the toolbox for type-2 fuzzy systems would be useful at least as a tool for developing initial ideas of a solution for a particular problem.

Index

Adaptive fuzzy inference system, 181, 218
Antecedent, 15, 16
Architecture,
For face recognition, 78, 82, 88, 90
For fingerprint recognition, 78, 81
For human recognition, 80
For voice recognition, 78, 81
Architecture optimization, 135, 138
Automation, 171, 179
Automated quality control, 171, 179, 181
Autonomous robotic systems, 203, 204, 206
Binary fuzzy relations, 12, 13
Biometric,
Information, 77, 78
Modalities, 78
Measures, 78, 81
Techniques, 79, 81
Templates, 78
Bisector, 17, 18

Centroid, 16, 18
Chromosome, 135, 136
Chromosome representation, 135, 136
Coding, 136
Complement, 8
Concentration, 14
Consequents, 14, 18
Containment, 7
Core, 6
Crossover, 7

Defuzzification, 16, 17, 18
Derivative freeness, 136
Dilation, 14
Direct encoding, 136
Divide and conquer principle, 79, 80

Edge border detection, 95, 96
Encoding schemes, 136, 211
Ensemble, 79
Evolutionary algorithms, 135, 137
Evolutionary optimization, 133, 134
Evolving,
Fuzzy systems, 133, 134
Network architecture, 78
Neural networks, 78, 79
Weights, 78
Extension principle, 11, 12

Face detection, 80, 87
Face recognition, 80, 87
Face recognition algorithms, 80, 87
Facial recognition, 81
Feature localization, 78
Fingerprint image, 81
Fingerprint recognition, 81
Fitness evaluation, 134
Footprint of uncertainty, 32, 33
Fractal dimension, 172, 180, 181
Fractal theory, 172, 180, 181
Fuzzy if-then rule, 14, 15
Fuzzy inference system,
Mamdani, 16, 17

Sugeno, 16, 23
Tsukamoto, 27, 28
Fuzzy integrals, 92, 93
Fuzzy modeling, 29, 30
Fuzzy reasoning, 11, 12
Fuzzy relations, 11
Fuzzy set, 6, 7
Fuzzy Sugeno integral, 92, 93
Fuzzy system for person
recognition, 80, 81

Gate of Experts, 79
Gasussian type-2 fuzzy set, 30, 31
Genetic algorithms, 208, 209
gλ-fuzzy measure, 92, 93

hierarchical mixture of
experts, 78, 79
hierarchical modular
architecture, 78, 79
human recognition, 81, 82
human recognition system, 81, 82

identification, 77, 78
integration module, 81, 91
intelligent control, 109, 110, 122, 134
intelligent manufacturing, 171, 172, 187
intersection,
type-1, 8
type-2, 36
interval type-2 fuzzy set, 32, 33
interval type-2 fuzzy system, 36

join operation, 34

largest of maximum, 18
linear combination, 23
linguistic variable, 13, 14
lower membership function, 33

Mamdani model, 16, 17
Max-min composition, 13
Mean of maximum, 18
Membership function, 8, 9
Gaussian, 9
Generalized bell, 9
Sigmoidal, 9
Trapezoidal, 9
Triangular, 8
Meet operation, 36
Minutia points, 78
Mixture model, 80
Mixture of experts, 80
Modular, 79
Modular approach, 77, 79, 80
Modular neural networks, 77, 84
Monitoring, 188, 189, 190
Mutation, 135, 136, 137

Non-singleton type-2 FLS, 41, 42
Normality, 7
Normalization method, 138

Objective function,
Operations of type-2
fuzzy sets, 34, 36
Optimization, 138
Optimization of
fuzzy systems, 138

Pattern recognition, 79, 87, 95
Pendubot, 161, 162
Plant monitoring, 188, 189
Population, 136
Primary methods of
Biometrics, 77, 78
Principle of Incompatibility, 13

Quality Control, 171, 172

Recognition system, 79, 80
Response integration, 80
Robotics, 203, 204, 205

Secondary membership, 31, 32, 33
Singleton type-2, 38, 39
Systematic design, 109, 112, 113
Smallest of maximum, 18
Soft computing, 1, 2, 3
Sound signal, 171, 179
Stable type-2 fuzzy control, 112, 113
Sugeno fuzzy measure, 92, 93
Sugeno model, 22, 23
Support, 7

Tracking, 203, 204
Type-1 fuzzy logic, 6, 7, 11

Type-2 fuzzy logic, 29, 30, 34, 37
Type-2 fuzzy logic
  integration, 93
Type-2 fuzzy rules, 37, 38
Type-2 fuzzy sets, 29
Type-2 Sugeno FLS, 41, 42

Union,
  Type-1, 7
  Type-2, 34
Upper membership function, 33

Voice data, 82
Voice recognition, 78, 82

Weight vector, 23
Weighted-sum, 23