

Ying Bai  
Hanqi Zhuang  
Dali Wang  
*Editors*



Advances in  
Industrial Control

# Advanced Fuzzy Logic Technologies in Industrial Applications



Springer

# Advances in Industrial Control

---

## ***Other titles published in this Series:***

*Digital Controller Implementation and Fragility*

Robert S.H. Istepanian and James F. Whidborne (Eds.)

*Optimisation of Industrial Processes at Supervisory Level*

Doris Sáez, Aldo Cipriano and Andrzej W. Ordys

*Robust Control of Diesel Ship Propulsion*

Nikolaos Xiros

*Hydraulic Servo-systems*

Mohieddine Jelali and Andreas Kroll

*Strategies for Feedback Linearisation*

Freddy Garces, Victor M. Becerra, Chandrasekhar Kambhampati and Kevin Warwick

*Robust Autonomous Guidance*

Alberto Isidori, Lorenzo Marconi and Andrea Serrani

*Dynamic Modelling of Gas Turbines*

Gennady G. Kulikov and Haydn A. Thompson (Eds.)

*Control of Fuel Cell Power Systems*

Jay T. Pukrushpan, Anna G. Stefanopoulou and Huei Peng

*Fuzzy Logic, Identification and Predictive Control*

Jairo Espinosa, Joos Vandewalle and Vincent Wertz

*Optimal Real-time Control of Sewer Networks*

Magdalene Marinaki and Markos Papageorgiou

*Process Modelling for Control*

Benoît Codrons

*Computational Intelligence in Time Series Forecasting*

Ajoy K. Palit and Dobrivoje Popovic

*Modelling and Control of mini-Flying Machines*

Pedro Castillo, Rogelio Lozano and Alejandro Dzul

*Rudder and Fin Ship Roll Stabilization*

Tristan Perez

*Hard Disk Drive Servo Systems* (2nd Ed.)

Ben M. Chen, Tong H. Lee, Kemao Peng and Venkatakrishnan Venkataramanan

*Measurement, Control, and Communication Using IEEE 1588*

John Eidson

*Piezoelectric Transducers for Vibration Control and Damping*

S.O. Reza Moheimani and Andrew J. Fleming

*Manufacturing Systems Control Design*

Stjepan Bogdan, Frank L. Lewis, Zdenko Kovačić and José Mireles Jr.

*Windup in Control*

Peter Hippe

*Nonlinear  $H_2/H_\infty$  Constrained Feedback Control*

Murad Abu-Khalaf, Jie Huang and Frank L. Lewis

*Practical Grey-box Process Identification*

Torsten Bohlin

*Modern Supervisory and Optimal Control*

Sandor Markon, Hajime Kita, Hiroshi Kise and Thomas Bartz-Beielstein

*Wind Turbine Control Systems*

Fernando D. Bianchi, Hernán De Battista and Ricardo J. Mantz

*Practical PID Control*

Antonio Vissioli

*Soft Sensors for Monitoring and Control of Industrial Processes*

Luigi Fortuna, Salvatore Graziani, Alessandro Rizzo and Maria Gabriella Xibilia

*Advanced Control of Industrial Processes*

Piotr Tatjewski

Publication due October 2006

---

Ying Bai, Hanqi Zhuang and Dali Wang (Eds.)

---

# **Advanced Fuzzy Logic Technologies in Industrial Applications**

---

With 220 Figures



**Springer**

Ying Bai, PhD  
Department of Computer Science  
and Engineering  
Johnson C. Smith University  
Charlotte, North Carolina  
USA

Hanqi Zhuang, PhD  
Department of Electrical Engineering  
Florida Atlantic University  
Boca Raton, Florida  
USA

Dali Wang, PhD  
Department of Physics, Computer Science  
and Engineering  
Christopher Newport University  
Newport News, Virginia  
USA

British Library Cataloguing in Publication Data  
Advanced fuzzy logic technologies in industrial  
applications. - (Advances in industrial control)  
1. Automatic control - Mathematics 2. Fuzzy logic  
I.Bai, Ying, 1956- II.Zhuang, Hanqi III.Wang, Dali  
629.8'01511313  
ISBN-13: 9781846284687  
ISBN-10: 1846284686

Library of Congress Control Number: 2006932288

Advances in Industrial Control series ISSN 1430-9491  
ISBN-10: 1-84628-468-6 e-ISBN 1-84628-469-4  
ISBN-13: 978-1-84628-468-7

Printed on acid-free paper

© Springer-Verlag London Limited 2006

MATLAB® and Simulink® are registered trademarks of The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, U.S.A. <http://www.mathworks.com>

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of 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 laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

9 8 7 6 5 4 3 2 1

Springer Science+Business Media  
[springer.com](http://springer.com)

# Advances in Industrial Control

## Series Editors

Professor Michael J. Grimble, Professor of Industrial Systems and Director  
Professor Michael A. Johnson, Professor (Emeritus) of Control Systems  
and Deputy Director

Industrial Control Centre  
Department of Electronic and Electrical Engineering  
University of Strathclyde  
Graham Hills Building  
50 George Street  
Glasgow G1 1QE  
United Kingdom

## Series Advisory Board

Professor E.F. Camacho  
Escuela Superior de Ingenieros  
Universidad de Sevilla  
Camino de los Descubrimientos s/n  
41092 Sevilla  
Spain

Professor S. Engell  
Lehrstuhl für Anlagensteuerungstechnik  
Fachbereich Chemietechnik  
Universität Dortmund  
44221 Dortmund  
Germany

Professor G. Goodwin  
Department of Electrical and Computer Engineering  
The University of Newcastle  
Callaghan  
NSW 2308  
Australia

Professor T.J. Harris  
Department of Chemical Engineering  
Queen's University  
Kingston, Ontario  
K7L 3N6  
Canada

Professor T.H. Lee  
Department of Electrical Engineering  
National University of Singapore  
4 Engineering Drive 3  
Singapore 117576

Professor Emeritus O.P. Malik  
Department of Electrical and Computer Engineering  
University of Calgary  
2500, University Drive, NW  
Calgary  
Alberta  
T2N 1N4  
Canada

Professor K.-F. Man  
Electronic Engineering Department  
City University of Hong Kong  
Tat Chee Avenue  
Kowloon  
Hong Kong

Professor G. Olsson  
Department of Industrial Electrical Engineering and Automation  
Lund Institute of Technology  
Box 118  
S-221 00 Lund  
Sweden

Professor A. Ray  
Pennsylvania State University  
Department of Mechanical Engineering  
0329 Reber Building  
University Park  
PA 16802  
USA

Professor D.E. Seborg  
Chemical Engineering  
3335 Engineering II  
University of California Santa Barbara  
Santa Barbara  
CA 93106  
USA

Doctor K.K. Tan  
Department of Electrical Engineering  
National University of Singapore  
4 Engineering Drive 3  
Singapore 117576

Professor Ikuo Yamamoto  
Kyushu University Graduate School  
Marine Technology Research and Development Program  
MARITEC, Headquarters, JAMSTEC  
2-15 Natsushima Yokosuka  
Kanagawa 237-0061  
Japan

---

## Series Editors' Foreword

The series *Advances in Industrial Control* aims to report and encourage technology transfer in control engineering. The rapid development of control technology has an impact on all areas of the control discipline. New theory, new controllers, actuators, sensors, new industrial processes, computer methods, new applications, new philosophies..., new challenges. Much of this development work resides in industrial reports, feasibility study papers and the reports of advanced collaborative projects. The series offers an opportunity for researchers to present an extended exposition of such new work in all aspects of industrial control for wider and rapid dissemination.

In the mid-1960s and contemporary with Kalman's pioneering papers on state-space models and optimal control, L.A. Zadeh began publishing papers on "fuzzy sets". It took another decade before the fuzzy-logic controller due to Mamdani and Assilion was reported in the literature (ca. 1974), and now the fuzzy-logic control paradigm is entering its fifth decade of development and application. Thus, this new *Advances in Industrial Control* monograph edited by Ying Bai, Hanqi Zhuang and Dali Wang on fuzzy-logic control and its practical application comes as a timely reminder of the wide range of problems that can be solved by this continually evolving methodology.

The volume can be considered in two parts, Chapters 1 to 5 cover a range of fundamental issues and then Chapters 6 to 21 range across a number of application studies. A cast of authors originating from the US, Mexico, Spain, P.R. China, Estonia, United Kingdom, Taiwan and Chile has contributed these 21 chapters. These diverse contributions have been melded together by editors Ying Bai, Hanqi Zhuang and Dali Wang to provide a systematic review of recent fuzzy-logic control applications.

The opening group of chapters examines a number of fundamental issues in constructing and applying fuzzy-logic controllers. The basic constructs of fuzzy-logic namely fuzzy sets, fuzzy rules and the defuzzification procedure are outlined in Chapter 2. Chapter 3 follows this basic material with a look at how fuzzy-logic controllers are implemented. A most interesting distinction between a static fuzzy-logic controller design and a dynamic (or adaptive) fuzzy-logic controller design is made in these chapters. This is one design route to enhancing the performance of fuzzy-logic controllers. Another route to performance enhancement is to extend the internal architecture of the fuzzy-logic controller by using "fine" and "coarse"

lookup tables and this modification is covered in Chapter 2. Also of considerable interest to the industrial engineer will be the material in Chapter 3 relating to the implementation of software and hardware fuzzy-logic controller solutions. Methods to ease the tuning of fuzzy-logic controllers are presented in Chapters 4 and 5. Tuning is a problem because fuzzy-logic controller performance can be tuned using scaled control gains, modifications to the membership functions, changes to the controller rule base or by adjusting entries to the inherent lookup table of the controller. None of these looks particularly simple and Chapters 4 and 5 make contributions to the tuning task.

A particular strength of this new volume in the *Advances in Industrial Control* series is the set of reports from a wide range of application studies, some 16 contributions in total, and these are found in Chapters 6 to 21. The subject areas are as follows:

*Manufacturing and Business Processes*

- Laser tracking system ~ Chapter 6
- Data mining ~ Chapter 17
- Manufacturing robot calibration ~ Chapter 20
- Plastic injection moulding ~ Chapter 21

*Image Processing*

- Visual image processing ~ Chapter 7
- Medical image processing ~ Chapter 8

*Vehicle Control Systems*

- Aspects of automobile control ~ Chapters 9, 10 and 11
- Mobile robot vehicles ~ Chapter 12
- Unmanned aerial vehicle control ~ Chapter 15 and 16

*Power System Control*

- Power distribution ~ Chapter 18
- Power generation ~ Chapter 19

This is quite a wide range of industrial areas and includes some non-control applications; for example, image processing and the survey chapter on fuzzy-logic applied to data-mining (Chapter 17). An interesting feature of the control problems is that some of them involve the control of “soft” variables like the distance between cars in the context of traffic control or the need for obstacle avoidance in the mobile-robotic-vehicle-control problem. These more relaxed control problems are thought to be more suited to the flexible approaches of fuzzy-logic control. Of the 16 application chapters, one is a survey chapter, eight demonstrate practical results using computer-simulated models and seven involve results from hardware prototypes, hardware experimental laboratory-scale rigs and some field trials. Consequently the volume has a healthy proportion of chapters with results from practical implementations of fuzzy-logic control.

This volume will be of considerable interest to all those involved in the development and application of the fuzzy-logic controller field. Industrial engineers and academic researchers should find the volume a useful indicator of the maturity of the fuzzy-logic controller paradigm and a valuable resource for exploring the potential of these controllers for industrial applications.

---

## Preface

Over the past four decades, the theory of Fuzzy Logic Control (FLC) and its applications have been widely developed and implemented in almost all aspects of our modern society, including education and research, manufacturing, medicine, and commercial applications. Application of the FLC technology leads to a revolution in modern control and intelligent control in our time. To help readers to understand and apply this technology in the real world, newer, updated information about fuzzy logic control and its applications is needed. This book intends to provide readers with advanced materials of the FLC technology applied to today's industrial settings. For this purpose, we invited authors, who are among the leaders in the related application fields, to contribute their experience and expertise to this book.

The book provides both adequate theoretical background and rich practical implementation examples on FLC in industrial applications. Compared with other similar books, up-to-date and advanced practical implementation of fuzzy onlogic controllers on in industrial applications, such as on-line dynamic control FLC, fuzzy interpolation methods, and various new tuning techniques, are presented. Many of these new techniques have been successfully tested in industrial environments, which can be critical for actual implementations, improve greatly performance of controllers, and save developers' time and energy.

The fuzzy inference technology discussed in most of the current books belongs to the category of static fuzzy inference systems, which means that both fuzzy membership functions and control rules are pre-determined prior to the implementation of a fuzzy logic controller for a plant. This approach is also called 'off-line' fuzzy inference system. The major disadvantage of this technique is that each output of a lookup table, derived from membership functions and control rules, is calculated based on a pre-assumed input range, which may not be accurate enough to real inputs as this fuzzy inference system is applied to an actual control system. Another drawback of this approach is that it makes a control parameter tuning process more difficult because of its 'off-line' nature.

To design a fuzzy logic controller that gives superior performance in a real system, the above-mentioned issue needs to be adequately addressed. To this end, a dynamic or 'on-line' fuzzy inference system is introduced and discussed in this

book. In the dynamic fuzzy inference system, membership functions and control rules are not determined until the fuzzy inference system is applied to a physical plant and each output of the lookup table is calculated based on the current inputs to the fuzzy inference system. This possesses the potential of not only greatly improving the control performance of the fuzzy inference system but also facilitating the parameter tuning process.

Another important contribution on the implementation of fuzzy control in the book is that some of new tuning techniques are discussed in a great detail. As everyone knows, the most difficult stage in a fuzzy control application is its tuning phase. How to efficiently tune a fuzzy control system and make it a perfect controller for a real system is a challenging task. Most existing books did not provide much information for such a tuning process. A few of books presented some, but those tuning techniques tend to be out of date and may not improve greatly the control performance of the system involved. New tuning methods, such as the  $\mu$ -law technique, the histogram equalization algorithm and the Bezier-based method are discussed in detail in this book. With these tuning techniques, a tuning process can be significantly simplified and the control performance of the system can be greatly improved.

Compared with other similar books on the subject, this book provides more up-to-date and advanced fuzzy control application techniques and examples in different fields, such as laser tracking and control, robot calibration, image processing and pattern recognition, medical engineering, autonomous underwater vehicles, audio systems, and data mining. These application examples will benefit practitioners from various industrial sectors.

The book is written to be easily understood by readers who may have no special knowledge in fuzzy logic and intelligent controls. The targeted readers of this book include a wide variety of people, from design engineers, application engineers and project managers working in industrial controls to undergraduate students, graduate students and faculty members in universities and scientific research laboratories.

Totally twenty chapters are included in this book. Chapter 1 provides basic knowledge of control technologies such as linear control, non-linear control and FLC. The fundamentals of FLC are presented in chapter 2. Chapters 3, 4 and 5 discuss the dynamic or 'on-line' fuzzy control systems and knowledge-based tuning procedures. Beginning in Chapter 6, various fuzzy control applications implemented in different industrial fields are provided. These application fields include:

- Laser tracking system
- Image processing and pattern recognition
- Medical engineering
- Guiding of the transports
- Automobiles control
- Autonomous mobile robots control
- Autonomous underwater vehicles control
- Fight control
- Path tracking and obstacles avoidance of UAVs

- Data mining control
- Power networks control
- Predictive control of a solar power plant
- Robots calibration
- Manufacturing welding systems control

Special thanks should be given to all chapter authors and those who have made valuable contributions to this book. This book could not be published without these indispensable contributions.

March 2006

Ying Bai  
Hanqi Zhuang  
Dali Wang

---

# Contents

List of Contributors .....	xi
<b>1 From Classical Control to Fuzzy Logic Control .....</b>	<b>1</b>
1.1 Introduction .....	1
1.2 Choice of Controllers .....	2
1.3 Control Problem .....	2
1.3.1 Motor Speed Control .....	2
1.3.2 Restrictions on Controller Inputs and Outputs .....	2
1.3.3 Nominal Plant .....	3
1.3.4 Perturbed Plants .....	3
1.3.5 Performance Criteria .....	3
1.3.6 Design Approach .....	4
1.4 Analysis and Synthesis of a Robust PID Controller .....	4
1.5 Analysis and Synthesis of a Robust Sliding Mode Controller .....	5
1.6 Analysis and Reconstruction of a Fuzzy Robust Controller .....	7
1.7 Conclusions .....	8
1.7.1 Comments on Controller Designs .....	8
1.7.2 From Classical Controls to Fuzzy Logic Controls .....	8
Appendix .....	9
References .....	15
<b>2 Fundamentals of Fuzzy Logic Control – Fuzzy Sets, Fuzzy Rules and Defuzzifications .....</b>	<b>17</b>
2.1 Introduction .....	17
2.2 Fuzzy Sets .....	19
2.2.1 Classical Sets and Operations .....	19
2.2.2 Mapping of Classical Sets to Functions .....	20
2.2.3 Fuzzy Sets and Operations .....	22
2.2.4 Comparison Between Classical and Fuzzy Sets .....	24
2.3 Fuzzification and Membership Functions .....	24
2.4 Fuzzy Control Rules .....	26
2.4.1 Fuzzy Mapping Rules .....	27

2.4.2	Fuzzy Implication Rules .....	28
2.5	Defuzzification and the Lookup Table.....	29
2.5.1	Mean of Maximum (MOM) Method .....	29
2.5.2	Center of Gravity (COG) Method.....	30
2.5.3	The Height Method (HM).....	30
2.5.4	The Lookup Table .....	31
2.5.5	Off-line and On-line Defuzzification.....	34
2.6	Architectures of Fuzzy Logic Controls.....	34
2.7	Summary.....	36
	References.....	36
<b>3</b>	<b>Implementation of Fuzzy Logic Control Systems .....</b>	<b>37</b>
3.1	Overview.....	37
3.2	Computational Complexity of Fuzzy Logic Control Algorithms.....	38
3.2.1	Fuzzification .....	39
3.2.2	Fuzzy Inference .....	40
3.2.3	Defuzzification .....	42
3.2.4	Static or Dynamic Fuzzy Logic Control.....	43
3.3	Software Solutions.....	44
3.3.1	General-Purpose Microprocessors .....	44
3.3.2	General-Purpose Microprocessors with Fuzzy Support .....	44
3.3.3	Software Development Tools for Fuzzy Logic System Design....	45
3.3.4	Online Implementation .....	46
3.3.5	Lookup Table Implementation .....	47
3.4	Hardware Solutions.....	48
3.4.1	Dedicated Fuzzy Processors .....	48
3.4.2	Fuzzy ASIC .....	50
3.5	Conclusions.....	51
	References.....	51
<b>4</b>	<b>Knowledge-based Tuning I: Design and Tuning of Fuzzy Control Surfaces with Bezier Function .....</b>	<b>53</b>
4.1	Introduction.....	53
4.2	Control Surfaces .....	54
4.3	Tuning with a Genetic Algorithm .....	59
4.4	Simulation Results .....	61
4.5	Conclusions.....	64
	References.....	64
<b>5</b>	<b>Knowledge-based Tuning II: <math>\mu</math>-Law Tuning of a Fuzzy Lookup Table.....</b>	<b>67</b>
5.1	Introduction.....	67
5.2	Gain and Surface Tuning of a Fuzzy Lookup Table .....	67
5.2.1	PD Controller and Fuzzy PD Controller.....	68
5.2.2	$\mu$ -Law Tuning of Fuzzy Lookup Table .....	69
5.3	Simulation and Experimental Results .....	74
5.4	Summary.....	80
	References.....	80

<b>6 Apply a Fuzzy Logic Controller to Suppress Noises and Coupling Effects for a Laser Tracking System</b> .....	83
6.1 Introduction.....	83
6.2 Operating Principle of LTS.....	85
6.3 The Control System, Coupling Effects and Nonlinearity .....	87
6.4 Design of Fuzzy Logic Controller .....	90
6.5 Simulation and Experimental Results.....	93
6.6 Conclusion.....	98
Acknowledgement.....	98
References.....	98
<b>7 Fuzzy Logic for Image Processing: Definition and Applications of a Fuzzy Image Processing Scheme</b> .....	101
7.1 Introduction.....	101
7.2 Fuzzy Image Processing Scheme Definition.....	102
7.2.1 Mapping the Image into the Fuzzify Domain .....	102
7.2.2 Operation on the Fuzzy Domain.....	104
7.2.3 Defuzzification Function .....	105
7.3 Examples of Fuzzy Image Processing .....	106
7.3.1 Fuzzy Binarization.....	106
7.3.2 Fuzzy Edge Definition.....	108
7.3.3 Fuzzy Geometry Measurement.....	110
7.4 Results and Conclusions .....	111
Acknowledgement.....	112
References.....	112
<b>8 Fuzzy Logic for Medical Engineering: An Application to Vessel Segmentation</b> .....	115
8.1 Introduction.....	115
8.2 Two Retinal Vessel Detection Procedures.....	117
8.3 Different Averages of a Fuzzy Set.....	117
8.4 The Heneghan Method with the Vorob'ev Fuzzy Set Average .....	122
8.5 The Zana-Klein Method and the Fuzzy Set Averages .....	123
8.6 Some Conclusions and Further Developments .....	124
Acknowledgement.....	127
References.....	127
<b>9 Fuzzy Logic for Transportation Guidance: Developing Fuzzy Controllers for Maintaining an Inter-Vehicle Safety Headway</b> .....	129
9.1 Introduction.....	129
9.2 The Autopia Program.....	130
9.2.1 Application of a Fuzzy Coprocessor to Automatic Vehicle Driving .....	131
9.2.2 The Longitudinal Control System .....	132
9.2.2.1 Design of the Fuzzy Cruise Control System .....	133
9.2.2.2 Fuzzy Adaptive Cruise Control Extension.....	135

9.3	Experiment Results.....	138
9.4	Conclusion.....	141
	Acknowledgement.....	141
	References .....	141
<b>10</b>	<b>Fuzzy Logic Control for Automobiles I: Knowledge-based Gear-position Decision.....</b>	<b>145</b>
10.1	Introduction.....	145
10.2	Knowledge-based GPD (KGPD) .....	146
10.2.1	Basic Ideas.....	146
10.2.2	Parameter Measurement and Normalization.....	149
10.2.3	Road Environment and Driver's Intention Recognition .....	150
10.2.4	Shift Schedules .....	152
10.2.5	Gear-Position Calculation.....	152
10.3	Experiment Test.....	153
10.4	Conclusion.....	155
	References.....	156
<b>11</b>	<b>Fuzzy Logic Control for Automobiles II: Navigation and Collision Avoidance System.....</b>	<b>159</b>
11.1	Introduction.....	159
11.2	Problem Formulation .....	160
11.3	Vehicle Navigation Control .....	161
11.4	Truck and Trailer Backing Control.....	164
11.5	Collision Avoidance System.....	165
11.5.1	State Transition Control.....	167
11.6	Results and Discussion .....	169
11.7	Conclusions.....	172
	Acknowledgement.....	172
	References.....	173
<b>12</b>	<b>Fuzzy Logic Based Control Mechanisms for Handling the Uncertainties Facing Mobile Robots in Changing Unstructured Environments .....</b>	<b>175</b>
12.1	Introduction.....	175
12.2	Type-2 Fuzzy Sets .....	177
12.2.1	Interval Type-2 Fuzzy Sets.....	179
12.2.2	Type-2 Set Theoretic Operations .....	180
12.3	Interval Type-2 Fuzzy Logic Controller (FLC) .....	181
12.3.1	The Fuzzifier .....	182
12.3.2	Rule Base .....	182
12.3.3	Fuzzy Inference Engine .....	182
12.3.4	Type-Reduction .....	183
12.3.5	Defuzzification .....	184
12.4	Experiments and Results.....	185
12.5	Conclusions.....	188
	References.....	189

<b>13 Combine Sliding Mode Control and Fuzzy Logic Control for Autonomous Underwater Vehicles</b> .....	191
13.1 Introduction.....	191
13.2 Autonomous Underwater Vehicles (AUVs) .....	193
13.3 Sliding Mode Controller .....	194
13.4 Fuzzy Sliding Mode Control (FSMC) .....	196
13.5 Sliding Mode Fuzzy Control (SMFC) .....	198
13.6 Sliding Mode Fuzzy Control Applied to AUVs.....	201
13.7 Conclusions.....	203
References.....	203
<b>14 Fuzzy Logic for Flight Control I: Nonlinear Optimal Control of Helicopter Using Fuzzy Gain Scheduling</b> .....	207
14.1 Introduction.....	207
14.2 The Nonlinear Dynamics of the Helicopter .....	208
14.3 Design of Linear Optimal Controller.....	211
14.4 Gain Scheduling Using Takagi-Sugeno Fuzzy Model.....	214
14.5 Conclusions.....	220
Acknowledgement.....	220
References.....	220
<b>15 Fuzzy Logic for Flight Control II: Fuzzy Logic Approach to Path Tracking and Obstacles Avoidance of UAVs</b> .....	223
15.1 Introduction.....	223
15.2 Strategy for Path Tracking and Obstacle Avoidance .....	224
15.3 Simulation Results .....	231
15.4 Conclusions.....	231
Acknowledgement.....	234
References.....	234
<b>16 Close Formation Flight Control of Multi-UAVs via Fuzzy Logic Technique</b> .....	237
16.1 Introduction.....	237
16.2 Flight Model.....	238
16.3 Fuzzy Logic Control .....	240
16.4 Simulations.....	244
16.5 Conclusion.....	246
Acknowledgement.....	247
References.....	247
<b>17 Applications of Fuzzy Logic in Data Mining Process</b> .....	249
17.1 Introduction.....	249
17.2 Foundation of Data Mining.....	250
17.3 Fuzzy Logic for Data Mining Basic.....	254
17.4 Algorithms of Fuzzy Logic in Data Mining Process.....	255
17.5 Conclusion.....	259

References.....	259
<b>18 Fuzzy Logic Control for Power Networks: A Multilayer Fuzzy Controller</b> .....	
18.1 Introduction.....	261
18.2 Description of Fuzzy Logic Systems .....	262
18.3 Functional Design of the Proposed Controller.....	263
18.4 Overall System Structure .....	268
18.5 Implementation and Test Results .....	269
18.5.1 The WSCC Test System.....	270
18.5.2 Four-generator Test System.....	273
18.6 Conclusion.....	276
References.....	277
<b>19 Fuzzy Predictive Control for Power Plants</b> .....	
19.1 Introduction.....	279
19.2 Model Predictive Control.....	280
19.3 Model Predictive Control for Power Plants .....	283
19.4 Fuzzy Control of Power Plants .....	284
19.5 Fuzzy Systems and Model Predictive Control.....	286
19.6 Integrating Fuzzy Systems and Model Prediction for Power Plant Control .....	287
19.7 Supervisory Fuzzy Predictive Control of a Boiler .....	289
19.8 Predictive Control of a Solar Power Plant with Fuzzy Goals and Constraints.....	291
19.9 Conclusions.....	294
Acknowledgement.....	295
References.....	295
<b>20 Fuzzy Logic for Robots Calibration – Using Fuzzy Interpolation Technique in Modeless Robot Calibration</b> .....	
20.1 Introduction.....	299
20.2 On-line 3D Fuzzy Interpolation .....	300
20.3 MATLAB Simulation Results .....	306
20.4 The Comparison Between the Modeless and Model-based Robots Calibration Methods .....	309
20.5 Conclusions.....	313
References.....	313
<b>21 Fuzzy Control on Manufacturing Welding Systems: To Apply Fuzzy Theory in the Control of Weld Line of Plastic Injection-molding</b> .....	
21.1 Introduction.....	315
21.2 Fuzzy Control Logic for the Weld Line Position.....	316
21.2.1 Fuzzification Interface.....	316
21.2.2 Production Rule .....	316
21.2.3 Decision Making Logic .....	317
21.2.4 Defuzzification Interface .....	318

21.3 Case Analyses.....	318
21.4 Simulation and Experimental Results .....	321
21.5 Conclusion.....	323
Acknowledgement.....	323
References .....	324
<b>Index.....</b>	<b>325</b>

---

## List of Contributors

*Chapter 1*

Charles P. Coleman

Dept of Aeronautics and Astronautics  
Massachusetts Institute of Technology  
Cambridge, MA 02139, USA  
e-mail: [ccoleman@mit.edu](mailto:ccoleman@mit.edu)

*Chapters 2, 3 and 20*

Ying Bai

Dept of Computer Science and Engineering  
Johnson C. Smith University  
Charlotte, NC 28216, USA  
e-mail: [ybai@jcsu.edu](mailto:ybai@jcsu.edu)

Dali Wang

Dept of Physics, Computer  
Science & Engineering  
Christopher Newport University  
Newport News, VA 23606, USA  
e-mail: [dwang@pcs.cnu.edu](mailto:dwang@pcs.cnu.edu)

*Chapter 4*

Hanqi Zhuang

Department of Electrical Engineering  
Florida Atlantic University  
Boca Raton, FL 33431  
e-mail: [zhuang@fau.edu](mailto:zhuang@fau.edu)

S. Wongsoontorn

Department of Electrical Engineering  
Florida Atlantic University  
Boca Raton, FL 33431

*Chapter 5*

Hanqi Zhuang  
Yingli Wang

Software Engineer  
3874 Corina Way  
Palo Alto, CA 94303  
e-mail: [victorw@gmail.com](mailto:victorw@gmail.com)

*Chapter 6*

Ying Bai  
Hanqi Zhuang  
Zvi S. Roth

Department of Electrical Engineering  
Florida Atlantic University  
Boca Raton, FL 33431  
e-mail: zroth@fau.edu

*Chapter 7*

Mario I. Chacón-Murguía

Graduate and Research Department  
Chihuahua Institute of Technology  
Chihuahua Chih., CP 31310 Mexico  
e-mail: mchacon@itchiuhhua.edu.mx

*Chapter 8*

Guillermo Ayala  
Teresa León  
Victoria Zapater

Dept. de Estadística e Investigación  
Operativa, Universidad de Valencia  
Avda. Vicent Andrés Estellés 1  
46100-Burjasot, Spain  
email: Guillermo.Ayala@uv.es  
email: Teresa.Leon@uv.es  
email: zapater\_mvi@gva.es

*Chapter 9*

José E. Naranjo  
Carlos González  
Ricardo García  
Teresa de Pedro

Instituto de Automática Industrial (CSIC)  
Ctra. Campo Real Km. 0,200  
La Poveda-Arganda del Rey  
28500, Madrid, Spain  
email: jnaranjo@iai.csic.es  
email: gonzalez@iai.csic.es  
email: ricardo@iai.csic.es  
email: tere@iai.csic.es

*Chapter 10*

Guihe Qin

Department of Computer Science and  
Technology  
Jilin University  
ChangChun 130025, P.R.China  
e-mail: ghqin@jlu.edu.cn

Anlin Ge

College of Automobile Engineering  
Jilin University  
ChangChun 130025, P.R.China  
e-mail: geal@jlu.edu.cn

Ju-Jang Lee

Department of Electrical Engineering and  
Computer Science  
Korea Advanced Institute of Science and  
Technology  
Deajeon 305-701, Korea

*Chapter 11*

Andri Riid  
Dmitri Pahhomov  
Ennu Rüstern

Department of Computer Control  
Tallinn University of Technology  
Ehitajate tee 5, Tallinn, 19086, Estonia  
email: andri@dcc.ttu.ee  
email: dmipah@hot.ee  
e-mail: ennu.rystern@dcc.ttu.ee

*Chapter 12*

Hani Hagras

Department of Computer Science  
The University of Essex  
Wivenhoe Park  
Colchester CO4 3SQ, England, UK  
e-mail: hani@essex.ac.uk

*Chapter 13*

Feijun Song

Department of Ocean Engineering  
Florida Atlantic University  
101 N. Beach Road  
Dania, FL 33004, USA  
e-mail: fsong@oe.fau.edu

Samuel M. Smith

Adpet Systems, Inc.  
2966 Fort Hill Road  
Eagle Mountain, UT 84043, USA  
e-mail: smithsm@samuelsmith.org

*Chapter 14*

Gwo-Ruey Yu

Department of Electrical Engineering  
National Ilan University  
1, Sec. 1, Shen-Lung Road  
Ilan City, Taiwan 260  
e-mail: gwoyu@niu.edu.tw

*Chapter 15*

Zhao Sun  
Ran Zhang

National Institute of Aerospace  
100 Exploration Way  
Hampton, VA 23666, USA  
e-mail: zhaosun@nianet.org  
e-mail: ranzhang@nianet.org

Tao Dong  
Xiaohong Liao  
David Y. Song

Electrical and Computer Engineering  
North Carolina A&T State University  
1601 E. Market Street  
Greensboro, NC 27411, USA  
e-mail: tdong@ncat.edu  
e-mail: xliao@ncat.edu  
e-mail: songyd@ncat.edu

*Chapter 16*

Yao Li

Institute for System Research  
Department of Electrical and Computer  
Engineering  
University of Maryland  
College Park, MD 20742  
e-mail: lee\_yao2003@yahoo.com

Zhao Sun

Ran Zhang

National Institute of Aerospace  
100 Exploration Way  
Hampton, VA 23666, USA  
e-mail: zhaosun@nianet.org  
e-mail: ranzhang@nianet.org

Bin Li

Liguo Weng

David Y. Song

Electrical and Computer Engineering  
North Carolina A&T State University  
1601 E. Market Street  
Greensboro, NC 27411, USA  
e-mail: libinncat@yahoo.com  
e-mail: liguo\_weng@hotmail.com  
e-mail: songyd@ncat.edu

*Chapter 17*

Hang Chen

Department of Computer Science and  
Engineering  
Johnson C. Smith University  
100 Beatties Ford Rd  
Charlotte, NC 28216 USA  
e-mail: hchen@jcsu.edu

*Chapter 18*

Ahmed Rubaai

Abdul R. Ofoli

Department of Electrical and Computer  
Engineering  
Howard University  
Washington DC 20059, USA  
e-mail: arubaai@howard.edu  
e-mail: ofoli@yahoo.com

*Chapter 19*

Aldo Z. Cipriano

Department of Electrical Engineering  
Pontificia Universidad Católica de Chile  
Vicuña Mackenna 4860, P. O. Box 306  
Santiago 6904411, Chile  
e-mail: aciprian@ing.puc.cl

*Chapter 21*

Mei-Yung Chen

Department of Industrial Education  
National Taiwan Normal University  
Taipei, 106, Taiwan, R. O. C.  
email: cmy@ntnu.edu.tw

Yi-Cheng Chen

Institute of Sports Equipment Technology  
Taipei Physical Education College  
Taipei, 111, Taiwan, R. O. C.  
email: ycchen@tpec.edu.tw

Shia-Chung Chen

Dept of Mechanical Engineering  
Chung Yuan Christian University  
Chung-Li, 320, Taiwan, R. O. C.  
email: shiachun@cycu.edu.tw

# From Classical Control to Fuzzy Logic Control

Charles P. Coleman

## 1.1 Introduction

Fuzzy logic controllers have been very successfully applied to many industrial applications over the past few decades [2-5]. However, the traditionally trained control engineer may be in need of a tutorial-like introduction to the comparison of classical control techniques and fuzzy logic control techniques in order either to transition to fuzzy logic control techniques, or to be convinced that fuzzy logic control techniques can be successfully applied to typical control problem well handled by classical control techniques.

It is the objective of this chapter to provide such a tutorial introduction. In this chapter, two standard and established classical control techniques, Proportional-Integral-Derivative (PID) control and sliding mode control, and fuzzy logic control, will be applied to a typical industrial control application – motor control. Each control technology will be applied to the control task at hand in the most equitable manner possible, so that resulting robustness performance can be compared.

The success of fuzzy logic controllers has prompted much comparison to classical control techniques, with detractors and supporters on both sides [6-11]. The ensuing arguments have often occurred over the similarities and synergies of classical and fuzzy logic approaches to control. Indeed, the practicing control engineer usually has little interest in ideological debates. It is in his or her interest to produce a working controller, with efficacy, making use of any and all techniques available.

It should be apparent by the end of this chapter that classical control techniques and fuzzy logic techniques can coexist and together bring great benefits to the practicing engineer by expanding the repertoire of available control techniques [12,13]. The following chapters in this edited volume will indeed clearly demonstrate that fuzzy logic control can provide great benefit to advanced industrial applications.

In the following sections we introduce the control problem. Then we synthesize robust controllers using two “classical” control techniques – PID and sliding mode control. Finally we synthesize a robust fuzzy logic controller for comparison. From the modeled performance of each of the controllers it will be apparent that classical and fuzzy logic control techniques can be applied to a typical industrial control application with similar performance results.

## 1.2 Choice of Controllers

In this chapter two classical control techniques, PID and sliding mode control, and fuzzy logic control are selected. PID control is examined regarding their use to synthesize robust controllers. PID control is chosen as one of the classical control techniques because it is one of the most commonly used controllers in industrial applications, and because it has plant perturbation robustness properties which can be mathematically analyzed. We choose sliding mode control as the second classical control technique because it is a robust nonlinear control method whose robustness properties can also be mathematically analyzed. Fuzzy logic control is chosen because it is the subject of this volume and, like the two classical control techniques, fuzzy logic control also has empirically demonstrated robustness properties [1]. In this chapter we do not present every robust control technique. Indeed, the popular and successful  $H_\infty$ , and adaptive control techniques are noticeably absent from this study. We restrict ourselves to fuzzy logic control and two other well established and popular control methods – PID and sliding mode control in order to keep this tutorial introduction brief, intelligible, and meaningful.

## 1.3 Control Problem

The control engineer is concerned about such properties as the robustness of a controller to plant perturbations and uncertainty. Taking the perspective of the control engineer who has several control tools available for the synthesis of robust controllers, we are interested in determining the ability of classical control and fuzzy logic control technique to synthesize a robust controller.

To assess the usefulness of fuzzy logic and classical control techniques in designing a robust controller, we choose a benchmark problem, and then compare the performance results of each control design method.

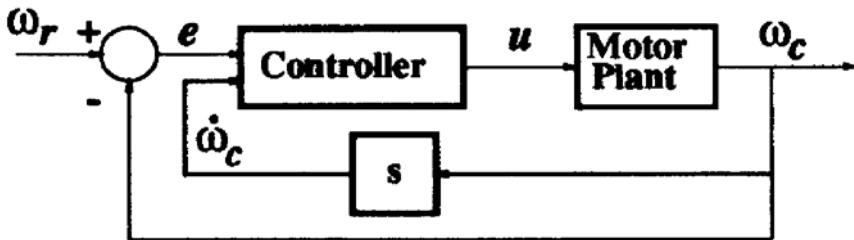
### 1.3.1 Motor Speed Control

Motivated by the work in [1, 14], we consider the typical control application of the robust speed control of a rotating motor, modeled by a third-order linear time-invariant transfer function. The general configuration for the motor speed control problem is given in Figure 1.1.

We use the nominal plant, perturbed motor plants, and the performance criteria given in [1] to design and compare robust fuzzy, PID, and sliding mode controllers.

### 1.3.2 Restrictions on Controller Inputs and Outputs

As shown in Figure 1.1, the reference step input speed  $\omega_r$ , and the output motor speed  $\omega_c$  are available for comparison. To keep the controller comparison fair, each controller only has access to the motor speed error,  $e = \omega_r - \omega_c$ , and the motor angular acceleration  $\dot{\omega}_c$  inputs. Each controller produces only one output  $u$ .



**Figure 1.1.** Motor speed control problem

### 1.3.3 Nominal Plant

The nominal motor plant is modeled by the following transfer function:

$$G_1(s) = \frac{5}{s(s+1)(s+2)} \quad (1.1)$$

### 1.3.4 Perturbed Plants

The two perturbed motor plants are modeled by the following transfer functions:

$$G_2(s) = \frac{15}{s(s+1)(s+2)} \quad (1.2)$$

$$G_3(s) = \frac{5}{s^2(s+2)} \quad (1.3)$$

### 1.3.5 Performance Criteria

The designed controllers must be robust to variations of system parameters. For the fuzzy logic, PID, and sliding control designs, one single controller must render acceptable the closed loop unit step responses of the nominal plant and the two perturbed plants. To be acceptable, the closed loop step responses must simultaneously have short rise time and no overshoot.

### 1.3.6 Design Approach

We design the fuzzy, PID, and sliding mode controllers based on the nominal plant given in Equation (1.1). Each designed controller is then applied to all three transfer functions, and numerical simulations are performed to analyze the controller's robustness.

In the next two sections, the design of the two classical controllers is presented: first PID and then sliding mode control. Then a fuzzy logic controller is synthesized for comparison to the classical control techniques. Plots of each controller's closed loop step response and control effort are shown in the Appendix to this chapter. All three control techniques, the two classical and the fuzzy control technique, will be found to provide adequate controllers, and for this application we can conclude that all three approaches should be available to practicing control engineer.

## 1.4 Analysis and Synthesis of a Robust PID Controller

The fuzzy logic controller designed in [1, 14] is similar to a nonlinear PD controller [15]. We design a linear PID controller to obtain satisfactory step response for all three transfer functions.

A linear PID controller can be characterized by the following transfer function:

$$C(s) = K_p + K_d s + \frac{K_i}{s} \quad (1.4)$$

Because, it is very difficult to realize a pure differentiator, we use the following transfer function for implementation:

$$C(S) = K_p + \frac{K_d s}{(\tau s + 1)} + \frac{K_i}{s} \quad (1.5)$$

To have a fair comparison, we use a small value for  $K_i = 0.001$ , just enough to keep steady state error zero. We use a small value for  $\tau = 0.01$ , so that its effect on the dynamics is minimal, but the derivative controller is still realizable.

The selection of  $K_p$  and  $K_d$  is based on the root locus of  $C(s)G_1(s)$ . The root locus gives us the locations of closed loop poles. The proportional and derivative gain values are chosen such that the closed loop pole locations are in the left half complex plain for all three plants. Another criterion for selection of these gains is the step response of the closed loop system.

We select the following gains:  $K_p = 2$ ,  $K_d = 5$ . With these gains, the closed loop system is stable for the open loop gain up to 51. Thus, in particular, the above PID controller stabilizes the systems given by the transfer functions in Equations (1.1) and (1.2). For the model given in Equation (1.3) we have closed loop stability for the open loop gain up to 30. The step responses have no overshoot and are critically damped as shown in Figures 1.2, 1.3 and 1.4.

## 1.5 Analysis and Synthesis of a Robust Sliding Mode Controller

To design a sliding mode controller, we convert the transfer function model into state space format. The controllable canonical form realization of the model given in Equation (1.1) is given by

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix} \quad (1.6)$$

$$y = [1 \ 0 \ 0]x$$

We consider the following sliding surface for this design (See [16] for details):

$$S = (\ddot{y} - \ddot{y}_d) + \lambda_1(\dot{y} - \dot{y}_d) + \lambda_2(y - y_d) \quad (1.7)$$

The choice of sliding surface is based on the following considerations:

- The relative degree of the system in Equation (1.6) with  $S$  as output should be 1. This ensures that the input  $u$  appears explicitly on the right hand side of the equation for  $\dot{S}$ .
- System dynamics on the sliding surface should be stable. This requires  $\lambda_1$ , and  $\lambda_2 > 0$ .

$y_d(t)$  specifies the reference trajectory to be tracked by the output. If one knows the entire trajectory of  $y_d$ , then the derivative information  $\dot{y}_d(t)$  and  $\ddot{y}_d(t)$  can be extracted offline and used in the design to improve the performance. If the information is not available, or as in this case,  $y_d(t)$  being unit step, is non-differentiable, then we can assume  $\dot{y}_d$  and  $\ddot{y}_d$  to be zero and let the robustness property of the sliding mode controller take care of the mismatch.

With the definition of sliding surface as above, we have reduced the design requirement from tracking  $y_d(t)$  to being on the surface  $S = 0$ . Once on the sliding surface, the dynamics (Equation (1.7)) are exponentially stable and asymptotic trajectory tracking is achieved. The control  $u$  is designed to make the surface  $S = 0$  attractive and to reach the surface in finite time.

Consider the following Lyapunov function:

$$V = \frac{1}{2}S^2 \quad (1.8)$$

Its time derivative is given by

$$\dot{V} = S\dot{S} \quad (1.9)$$

As the system has relative degree 1 with  $S$  as output, we can solve for  $u$  from the equation

$$S = -K \operatorname{sgn}(S) \quad (1.10)$$

This results in a negative definite  $\dot{V}$  and also guarantees finite time convergence to the sliding surface. However, the controller will result in high frequency chattering near sliding surface. To avoid this, we use the following expression to solve for  $u$ <sup>1</sup>:

$$\dot{S} = -KS \quad (1.11)$$

$$\Rightarrow \dot{V} = -KS^2 \quad (1.12)$$

Expanding both sides of the equation, we get

$$\begin{aligned} \dot{S} &= -2x_2 - 3x_3 + 5u + \lambda_1 x_3 + \lambda_2 x_2 \\ &= -K(x_3 + \lambda_1 x_2 + \lambda_2 x_1 - \lambda_2 y_d) \end{aligned}$$

The  $u$  obtained from this equation will be used for the control.

This controller needs access to all three states, whereas the fuzzy logic controller of [1, 14] makes use of only  $x_1$  and  $x_2$ . To have a fair comparison, we will construct an observer to get an estimate of the state variable  $x_3$ .

A standard Luenburger observer is designed to get an estimate of  $x_3$ . We use the state space model of the plant given in Equation (1.6) to design the observer as follows:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + bu + L(y - \hat{y}) \quad (1.13) \\ \hat{y} &= cx \end{aligned}$$

where, the matrices  $A$ ,  $b$  and  $c$  correspond to the plant model in Equation (1.6). The matrix  $L$  is chosen such that the eigenvalues of observer error dynamics  $(A - LC)$  are in the left half plane and the error dynamics is faster than the dynamics on the sliding surface.

The closed loop step responses with this sliding mode controller are shown in Figures 1.2, 1.3 and 1.4. The responses have no overshoot and are critical. The controller is robust to input gain changes as well as to the changes in system dynamics while keeping the input magnitude small.

Note: With the use of Equation (1.11), the controller is essentially a linear state feedback controller. It results in closed loop eigenvalues of

---

<sup>1</sup>With this controller, the trajectory is not guaranteed to reach the sliding surface in finite time.

$\{-1.3392, -4.33 \pm 1.91j\}$  for the plant model in Equation (1.1). The same controller, when applied to other plants, will give the following closed loop eigenvalue locations: for the perturbed model given by Equation (1.2)  $\{-1.32, -3.55, -19.12\}$ , and for the perturbed model given by Equation (1.3)  $\{-1.4, -3.79 \pm 2.62j\}$ . Thus, by design, the increase in gain makes the closed loop system more stable.

## 1.6 Analysis and Reconstruction of a Fuzzy Robust Controller

We implement the robust fuzzy controller given in [1,14]. The inputs to the fuzzy controller are  $e$  and  $\dot{\omega}_c$ . The output of the fuzzy controller is  $u$ . The universes of discourse of  $e$ ,  $\dot{\omega}_c$ , and  $u$  are partitioned into seven fuzzy sets:

- NB - negative big
- NM - negative medium
- NS - negative small
- ZE - zero
- PS - positive small
- PM - positive medium
- PB - positive big

Each fuzzy set is represented by a Gaussian membership function. The rule base of the fuzzy logic controller contains 49 rules that are tabulated in Table 1.1. The output of each rule is determined by min-inference. The crisp output  $u$  of the fuzzy logic controller is generated by centroid defuzzification.

**Table 1.1** Fuzzy logic controller rule base

		$e$						
		NB	NM	NS	ZE	PS	PM	PB
$\omega_c$	PB	NB	NB	NB	NB	NM	NS	ZE
	PM	NB	NB	NB	NM	NS	ZE	PS
	PS	NB	NB	NM	NS	ZE	PS	PM
	ZE	NB	NM	NS	ZE	PS	PM	PB
	NS	NM	NS	ZE	PS	PM	PB	PB
	NM	NS	ZE	PS	PM	PB	PB	PB
	NB	ZE	PS	PM	PB	PB	PB	PB

The step response of this fuzzy logic controller applied to the nominal and perturbed plants is shown in Figures 1.2, 1.3 and 1.4. The step responses for all three plants have short rise times and no overshoot. Thus, they meet the specified robust performance criteria. The control effort generated by the fuzzy logic controller is also shown in the Appendix to this chapter, as are the control efforts for the PID and sliding mode controllers (Figures 1.5, 1.6 and 1.7).

## 1.7 Conclusions

### 1.7.1 Comments on Controller Designs

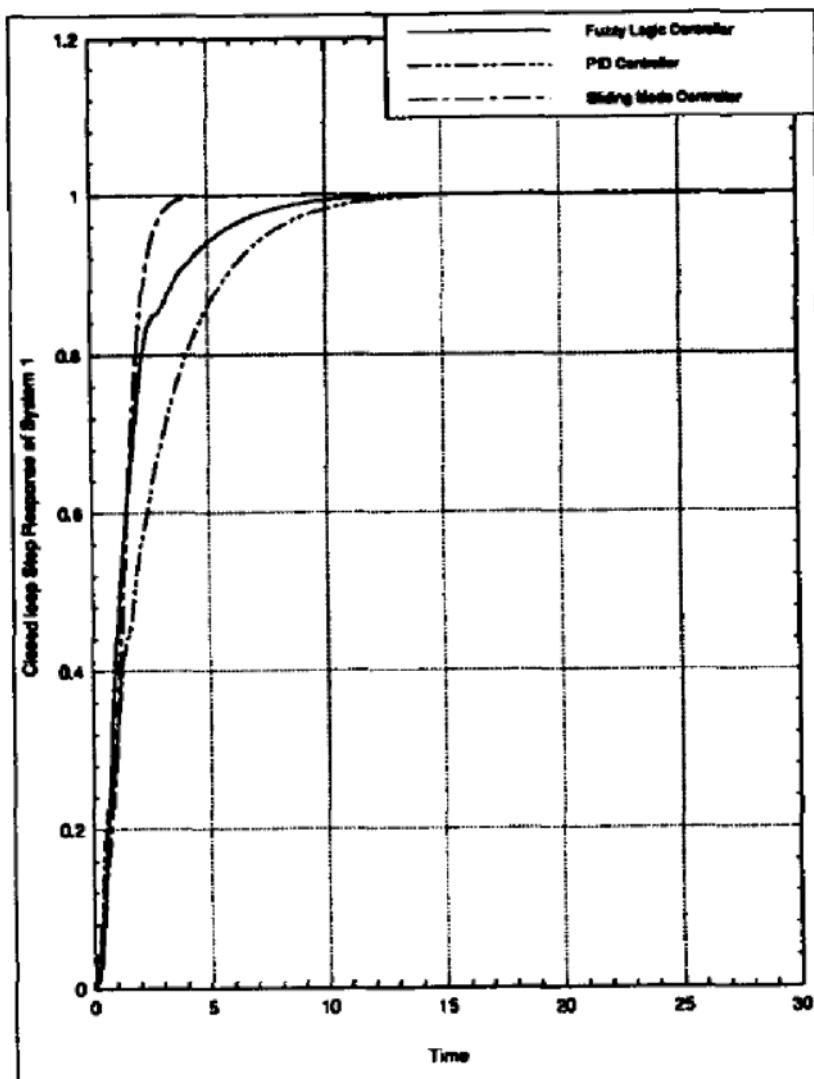
The control engineer proficient in PID and sliding mode control techniques can readily synthesize robust controllers to perform the benchmark control task presented in this chapter. As demonstrated in [I] and verified in this chapter, the control engineer utilizing fuzzy control techniques can readily achieve the same goal. Thus, we have shown that classical and fuzzy logic control tools are available to the control engineer for use in synthesizing an acceptable robust controller, a typical motor control problem.

Fuzzy logic control software tools are readily available and so the developing of a controller for the benchmark motor control problem in this chapter for this problem was relatively simple to implement and it provided satisfactory results. Software tools are also readily available for the design and simulation of the PID and sliding mode controllers for these problems, and so these two controllers were also easy to implement.

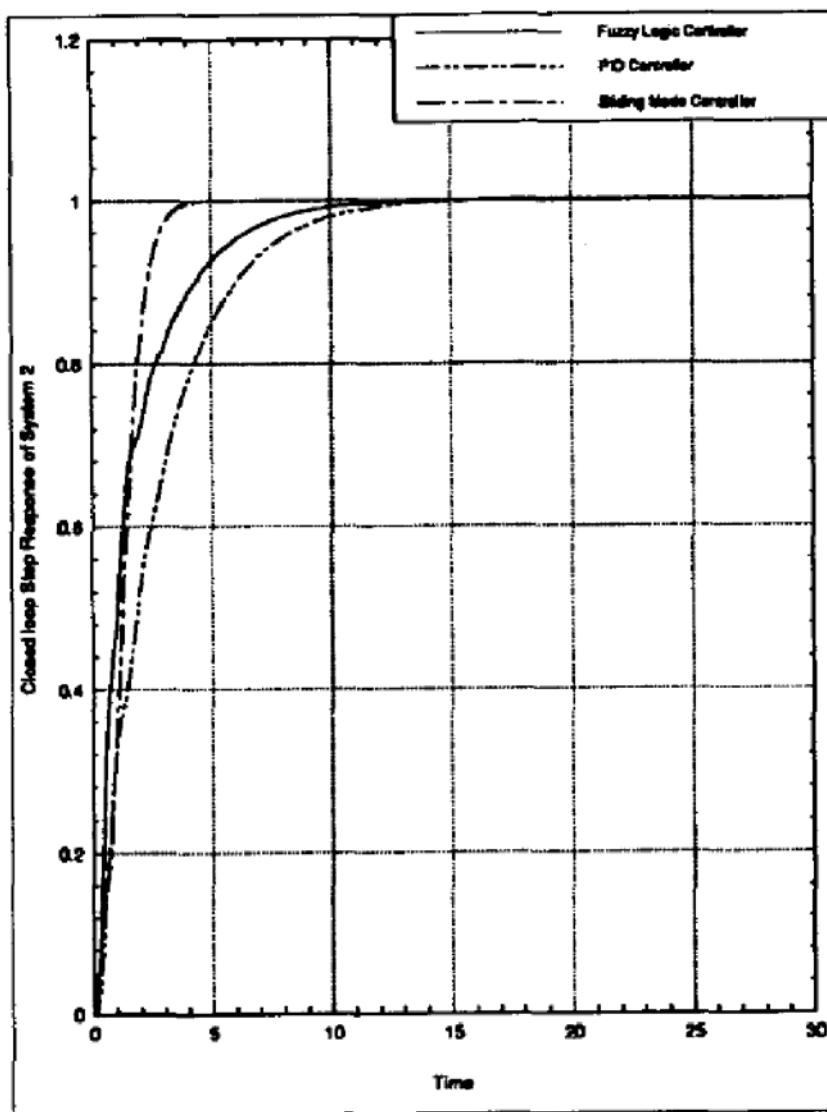
### 1.7.2 From Classical Controls to Fuzzy Logic Controls

Hopefully, this introductory tutorial of PID, sliding mode control, and fuzzy logic control of motor speed clearly demonstrates that fuzzy logic control should have a place in the industrial control engineer's toolbox alongside classical control techniques. Classical control should not be abandoned but, as other chapters in this volume demonstrate, the control engineer should take advantage of the power of fuzzy logic control where he or she can in advanced industrial control applications. We encourage the continued development of fuzzy logic control techniques and its integration and synergistic development with classical control techniques in the future.

## Appendix. Controller Performance Results



**Figure 1.2.** Closed loop step response of nominal plant  $G_I(s)$



**Figure 1.3.** Closed loop step response of the perturbed model  $G_2(s)$

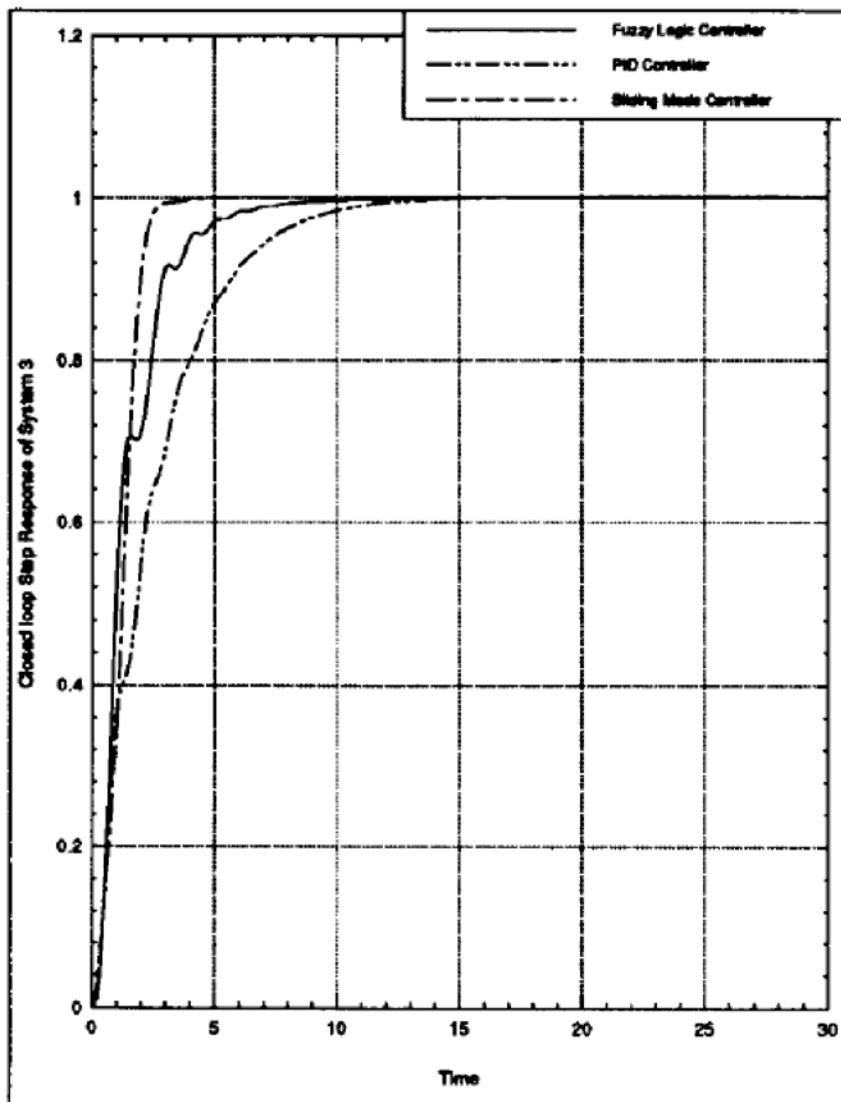
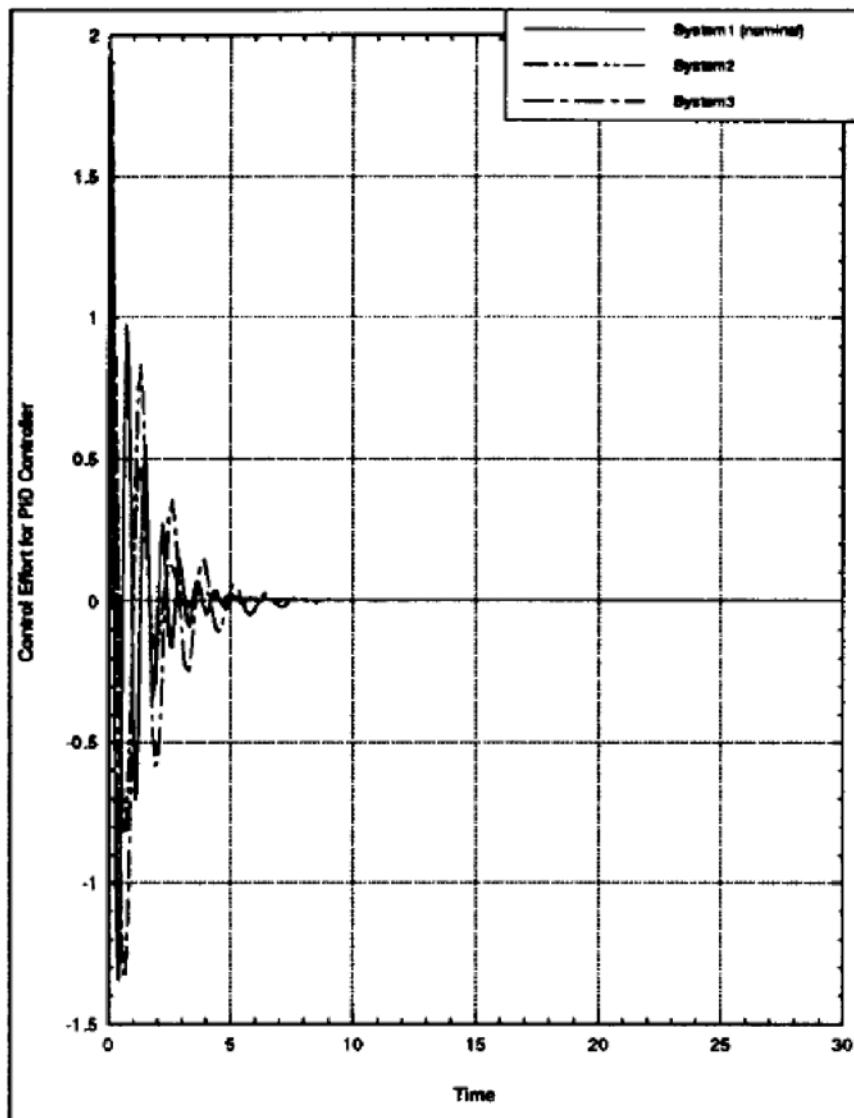


Figure 1.4. Closed loop step response of the perturbed model  $G_3(s)$



**Figure 1.5.** PID controller control effort

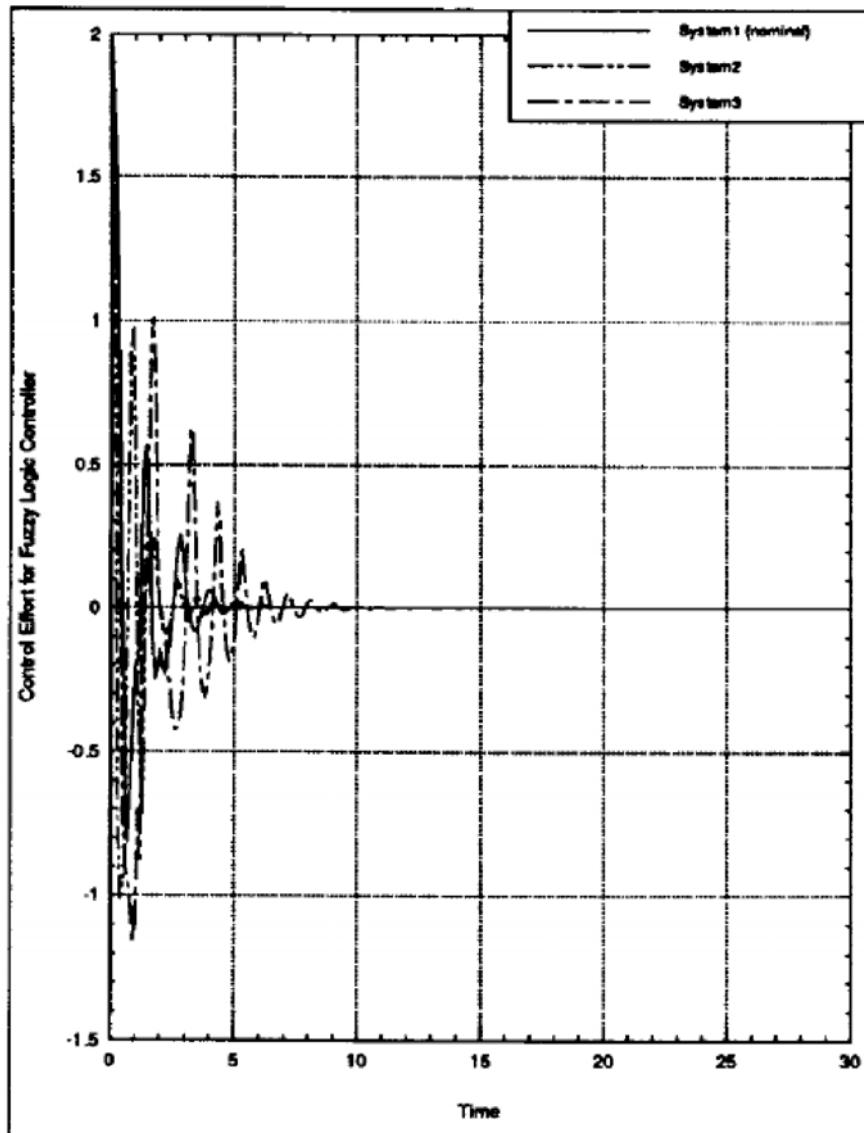


Figure 1.6. Fuzzy logic controller control effort

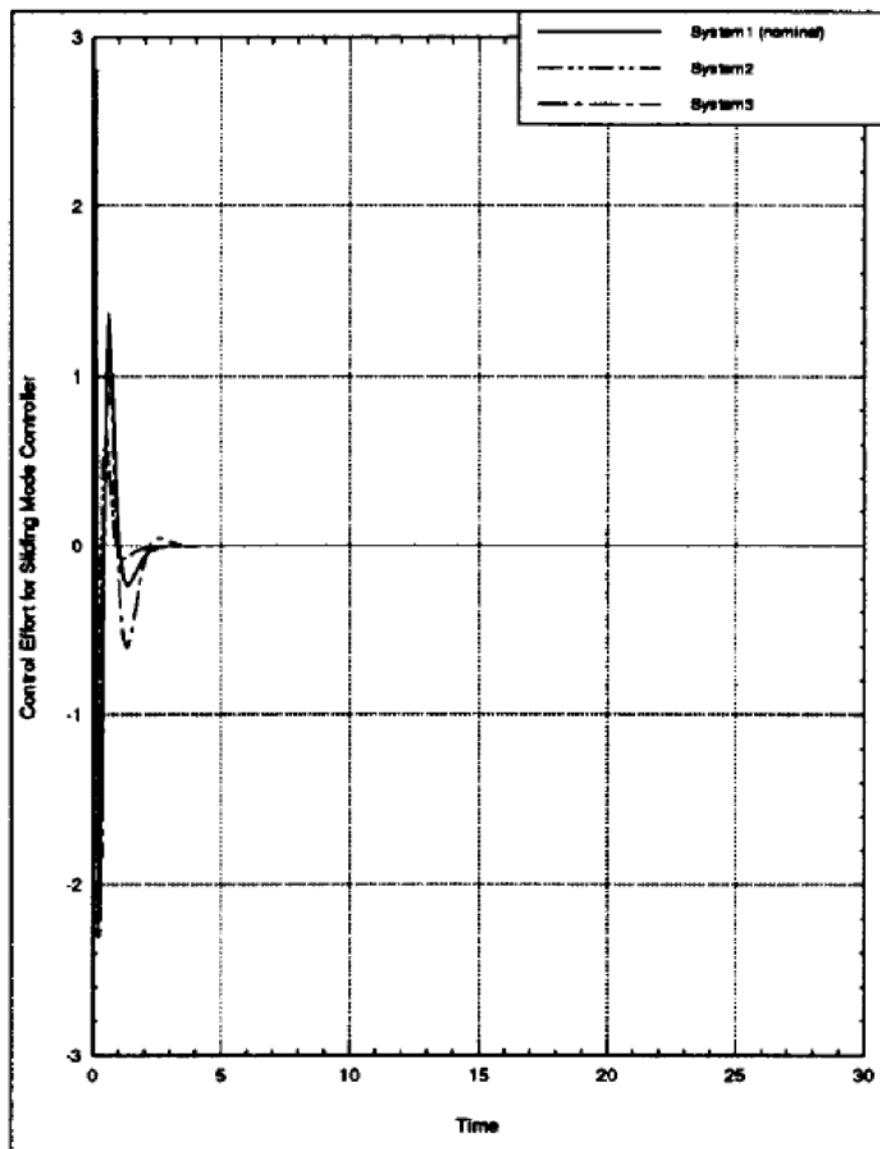


Figure 1.7. Sliding mode controller control effort

## References

- [1] H. T. Nguyen, C.-W. Tao, W. E. Thompson, "An Empirical Study of Robustness of Fuzzy Systems", *Proc. of 2nd IEEE Intl. Conf. on Fuzzy Systems*, pp. 1340-1345.
- [2] R. M. Tong, "An Annotated Bibliography of Fuzzy Control", in *Industrial Applications of Fuzzy Control*, M. Sugeno, Ed., North- Holland, Amsterdam, Holland, 1985, pp. 249-269.
- [3] T. Terano, K. Asai, M. Sugeno. *Fuzzy Systems Theory and Its Applications*. Academic Press, Boston, MA, USA.
- [4] W. Pedrycz. *Fuzzy Control and Fuzzy Systems*, 2nd extended ed. Wiley, New York, NY, USA.
- [5] H. Berenji, "Fuzzy Logic Controllers", in *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, R. R. Yager and L. A. Zadeh, Eds., Kluwer Academic Publishers, Boston, MA, USA, 1992, pp. 69-96.
- [6] S. Chiu, S. Chand, D. Moore, A. Chaudhary, "Fuzzy Logic for Control of Roll and Moment for a Flexible Wing Aircraft", *IEEE Control Systems Magazine*, pp. 42-48, Vol. 11, No. 4, June 1991.
- [7] A. L. Schwartz, "Comments on Fuzzy Logic Control of Roll and Moment for a Flexible Wing Aircraft", *IEEE Control Systems Magazine*, pp. 61-62, Vol. 12, No. 1, February 1992.
- [8] S. Chiu, "Author's Reply", *IEEE Control Systems Magazine*, pp. 62-63, Vol. 12, No. 1, February 1992.
- [9] E. Cox, "Adaptive Fuzzy Systems", *IEEE Spectrum*, pp 27-31, Vol. 30, No. 2, February 1993.
- [10] C. J. Herget, Ed., "Reader's Forum", *IEEE Control Systems Magazine*, pp. 5-7, Vol. 13, No. 3, June 1993.
- [11] M. Athans, "Control - The Adventure Continues", Bode Lecture, *32nd IEEE Conference on Decision and Control*, San Antonio, TX, USA, December 15-17, 1993.
- [12] E. H. Mamdani, "Twenty Years of Fuzzy Control: Experiences Gained and Lessons Learnt", *Proc. of 2nd IEEE Intl. Conf. on Fuzzy Systems*, pp. 339-344.
- [13] M. Tomizuka, "Fuzzy Control in Control Engineer's Tool Box", Lecture, *ARO/NASA Workshop on Formal Models for Intelligent Control*, MIT, Cambridge, MA 02139, USA, September 30 - October 2, 1993.
- [14] C.-W. Tao, R. Mamlook, W. E. Thompson, "Reduction of Complexity for a Robust Fuzzy Controller", *Proc. of 2nd IEEE Intl. Conf. on Fuzzy Systems*, pp. 1346-1349.
- [15] H. Ying, W. Siler, J. J. Buckley, "Fuzzy Control Theory: A Nonlinear Case", *Automatica*, Vol. 26, No. 3, pp. 513-420, 1990.
- [16] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 2nd ed., 1991.

## **Fundamentals of Fuzzy Logic Control – Fuzzy Sets, Fuzzy Rules and Defuzzifications**

Ying Bai and Dali Wang

### **2.1 Introduction**

Fuzzy logic idea is similar to the human being's feeling and inference process. Unlike classical control strategy, which is a point-to-point control, fuzzy logic control is a range-to-point or range-to-range control. The output of a fuzzy controller is derived from fuzzifications of both inputs and outputs using the associated membership functions. A crisp input will be converted to the different members of the associated membership functions based on its value. From this point of view, the output of a fuzzy logic controller is based on its memberships of the different membership functions, which can be considered as a range of inputs.

Fuzzy ideas and fuzzy logic are so often utilized in our routine life that nobody even pays attention to them. For instance, to answer some questions in certain surveys, most time one could answer with 'Not Very Satisfied' or 'Quite Satisfied', which are also fuzzy or ambiguous answers. Exactly to what degree is one satisfied or dissatisfied with some service or product for those surveys? These vague answers can only be created and implemented by human beings, but not machines. Is it possible for a computer to answer those survey questions directly as a human beings did? It is absolutely impossible. Computers can only understand either '0' or '1', and 'HIGH' or 'LOW'. Those data are called crisp or classic data and can be processed by all machines.

Is it possible to allow computers to handle those ambiguous data with the help of a human being? If so, how can computers and machines handle those vague data? The answer to the first question is yes. But to answer the second question, we need some fuzzy logic techniques and knowledge of fuzzy inference system.

The idea of fuzzy logic was invented by Professor L. A. Zadeh of the University of California at Berkeley in 1965 [1]. This invention was not well recognized until Dr. E. H. Mamdani, who is a professor at London University, applied the fuzzy logic in a practical application to control an automatic steam engine in 1974 [2], which is almost ten years after the fuzzy theory was invented. Then, in 1976, Blue Circle Cement and SIRA in Denmark developed an industrial application to control cement kilns [3]. That system began to operation in 1982. More and more fuzzy implementations have been reported since the 1980s, including those applications in industrial manufacturing, automatic control,

automobile production, banks, hospitals, libraries and academic education. Fuzzy logic techniques have been widely applied in all aspects in today's society.

To implement fuzzy logic technique to a real application requires the following three steps:

1. Fuzzification – convert classical data or crisp data into fuzzy data or Membership Functions (MFs)
2. Fuzzy Inference Process – combine membership functions with the control rules to derive the fuzzy output
3. Defuzzification – use different methods to calculate each associated output and put them into a table: the lookup table. Pick up the output from the lookup table based on the current input during an application

As mentioned before, all machines can process crisp or classical data such as either '0' or '1'. In order to enable machines to handle vague language input such as 'Somehow Satisfied', the crisp input and output must be converted to linguistic variables with fuzzy components. For instance, to control an air conditioner system, the input temperature and the output control variables must be converted to the associated linguistic variables such as 'HIGH', 'MEDIUM', 'LOW' and 'FAST', 'MEDIUM' or 'SLOW'. The former is corresponding to the input temperature and the latter is associated with the rotation speed of the operating motor. Besides those conversions, both the input and the output must also be converted from crisp data to fuzzy data. All of these jobs are performed by the first step – fuzzification.

In the second step, to begin the fuzzy inference process, one need combine the Membership Functions with the control rules to derive the control output, and arrange those outputs into a table called the lookup table. The control rule is the core of the fuzzy inference process, and those rules are directly related to a human being's intuition and feeling. For example, still in the air conditioner control system, if the temperature is too high, the heater should be turned off, or the heat driving motor should be slowed down, which is a human being's intuition or common sense. Different methods such as Center of Gravity (COG) or Mean of Maximum (MOM) are utilized to calculate the associated control output, and each control output should be arranged into a table called lookup table.

During an actual application, a control output should be selected from the lookup table developed from the last step based on the current input. Furthermore, that control output should be converted from the linguistic variable back to the crisp variable and output to the control operator. This process is called defuzzification or step 3.

In most cases the input variables are more than one dimension for real applications, and one needs to perform fuzzification or develop a Membership Function for each dimensional variable separately. Perform the same operation if the system has multiple output variables.

Summarily, a fuzzy process is a process of crisp-fuzzy-crisp for a real system. The original input and the terminal output must be crisp variables, but the intermediate process is a fuzzy inference process. The reason why one needs to change a crisp to a fuzzy variable is that, from the point of view of fuzzy control or a human being's intuition, no absolutely crisp variable is existed in our real world.

Any physical variable may contain some other components. For instance, if someone says: the temperature here is high. This high temperature contains some middle and even low temperature components. From this point of view, fuzzy control uses universal or global components, not just a limited range of components as the classical variables did.

With the rapid development of fuzzy technologies, different fuzzy control strategies have been developed based on different classical control methods, such as PID-fuzzy control [4], sliding-mode fuzzy control [5], neural fuzzy control, adaptor fuzzy control [6] and phase-plan mapping fuzzy control [7]. More and more new fuzzy control strategies or combined crisp and fuzzy control techniques are being developed and will be applied to many areas in our society in the future.

This chapter starts with an introduction to fuzzy logic control and fuzzy terminology. Following sections provide detailed description of the fuzzy inference process and the architecture of fuzzy logic control. Section 2 discusses fuzzy sets and crisp sets. Fuzzifications and the membership function are covered in Section 3. Fuzzy control rules and defuzzification are discussed in Sections 4 and 5, respectively. Some related combined fuzzy control methods are provided in the remaining sections of this chapter.

## 2.2 Fuzzy Sets

The concept of the fuzzy set is only an extension of the concept of a classical or crisp set. The fuzzy set is actually a fundamentally broader set compared with the classical or crisp set. The classical set only considers a limited number of degrees of membership such as '0' or '1', or a range of data with limited degrees of membership. For instance, if a temperature is defined as a crisp high, its range must be between 80 °F and higher and it has nothing to do with 70 °F or even 60 °F. But the fuzzy set will take care of a much broader range for this high temperature. In other words, the fuzzy set will consider a much larger temperature range such as from 0 °F to higher degrees as a high temperature. The exact degree to which the 0 °F can contribute to that high temperature depends on the membership function. This means that the fuzzy set uses a universe of discourse as its base and it considers an infinite number of degrees of membership in a set. In this way, the classical or crisp set can be considered as a subset of the fuzzy set.

### 2.2.1 Classical Sets and Operations

A classical set is a collection of objects in a given range with a sharp boundary. An object can either belong to the set or not belong to the set. For example, we assume to create a faculty set or a faculty collection  $F$  with ten-faculty members  $x_1, x_2, \dots, x_{10}$ , in a college:

$$F = \{ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \} \quad (2.1)$$

In general, the entire object of discussion  $F$  is called a universe of discourse, and each member  $x_i$  is called an element. Assuming that elements  $x_1 \sim x_4$  belong to the department of computer science, which can be considered as another set  $A$ . The elements  $x_1 \sim x_3$  are under age 40, which can be considered as a set  $B$ . Therefore the following relations exist:

$$\begin{aligned} X &\in F \\ A &= \{x_1, x_2, x_3, x_4\} \in F \\ B &= \{x_1, x_2, x_3\} \subset A \end{aligned} \quad (2.2)$$

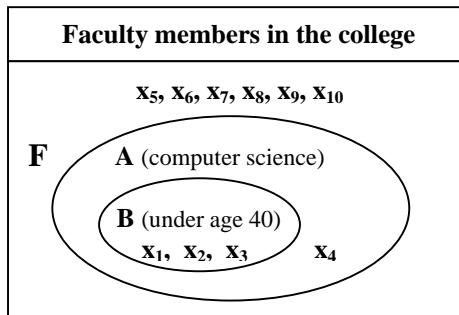
It can be seen that all elements in set  $B$  belong to set  $A$ , or the set  $A$  contains set  $B$ . In this case, set  $B$  can be considered as a subset of set  $A$  and can be expressed as  $B \subset A$ .

The relationship among different sets we discussed above can be described in Figure 2.1. The basic classical set operations include complement, intersection and union, which are represented as

$$\text{Complement of } A (A^C) \quad A^C(x) = 1 - A \quad (2.3)$$

$$\text{Intersection of } A \text{ and } B (A \cap B) \quad A \cap B = A(x) \cap B(x) \quad (2.4)$$

$$\text{Union of } A \text{ and } B (A \cup B) \quad A \cup B = A(x) \cup B(x) \quad (2.5)$$



**Figure 2.1.** The classical set of faculty members in a college

The representations of those classical set operations are shown in Figure 2.2.

It is clear that an element either belongs to a set or does not belong to that set in the classical set and its operation. There is a sharp boundary between different elements for different sets and these cannot be mixed with each other. However, for the fuzzy set, it has different laws.

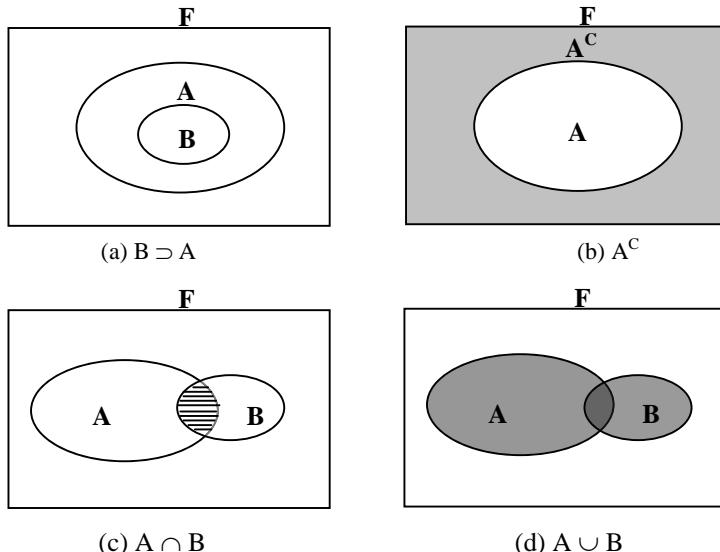
## 2.2.2 Mapping of Classical Sets to Functions

The classical set we discussed in the last section can be represented or mapped into some functions. This means that we can relate set-theoretic forms to function-theoretic representations, and map elements on one universe of discourse to elements or sets in another universe. For the classical or crisp sets, this mapping is

very easy and straightforward. Assume that  $X$  and  $Y$  are two different universes of discourse. If an element  $x$  belongs to  $X$  and it corresponds to an element  $y$  belonging to  $Y$ , the mapping between them can be expressed as

$$\mu_A(x) = \begin{cases} 1, & (x \in A) \\ 0, & (x \notin A) \end{cases} \quad (2.6)$$

where  $\mu_A$  represents 'membership' in a set  $A$  for the element  $x$  in the universe. This membership idea is exactly a mapping from the element  $x$  in the universe  $X$  to one of two elements in universe  $Y$ , or to elements 0 or 1.



**Figure 2.2.** The classical set and its operations

Still using the example in the last section, a set of faculty members in a college that contains two subsets  $A$  and  $B$ , we want to define a new set  $P(x)$  as

$$P(x) = \{ A, B \} \quad (2.7)$$

and mapping these two sets from universe of discourse  $F$  to another universe  $Y$  with two elements, 0 or 1.

$$M(P(x)) = \{ (1, 1, 1, 1, 0, 0, 0, 0, 0, 0), (1, 1, 1, 0, 0, 0, 0, 0, 0, 0) \} \quad (2.8)$$

Now we can define operations of two classical sets, says  $A$  and  $B$ , on the universe  $F$  as follows:

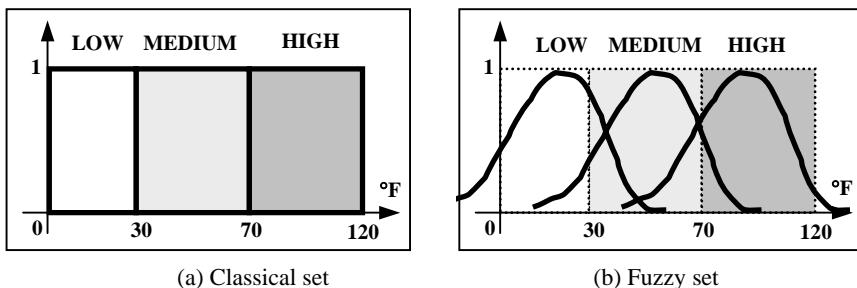
$$\text{Union: } A \cup B = \mu_A(x) \cup \mu_B(x) = \max(\mu_A(x), \mu_B(x))$$

$$\begin{aligned} \text{Intersection: } & A \cap B = \mu_A(x) \cap \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \\ \text{Complement: } & A^C = F \setminus A \end{aligned} \quad (2.9)$$

### 2.2.3 Fuzzy Sets and Operations

As we discussed in the previous section, the classical set has a sharp boundary, which means that a member either belongs to that set or does not. Also, this classical set can be mapped to a function with two elements, 0 or 1. For example, in the previous section, we defined the faculty member in the department of computer science as set  $A$ . A faculty either fully belongs to this set ( $\mu_A(x) = 1$ ) if one is a faculty of the computer science department or has nothing to do with set  $A$  ( $\mu_A(x) = 0$ ) if he is not a faculty in that department. This mapping is straightforward with sharp boundary without any ambiguity. In other words, this ‘fully belonging to’ can be mapped as a member of set  $A$  with degree of 1, and ‘not belong to’ can be mapped as a member of set  $A$  with degree of 0. This mapping is similar to a black-and-white binary categorization.

Compared with a classical set, a fuzzy set allows members to have a smooth boundary. In other words, a fuzzy set allows a member to belong to a set to some partial degree. For instance, still using the temperature as an example, the temperature can be divided into three categories: LOW (0 ~ 30 °F), MEDIUM (30 °F ~ 70 °F) and HIGH (70 ~ 120 °F) from the point of view of the classical set, which is shown in Figure 2.3a.



**Figure 2.3.** Representations of classical and fuzzy sets

In the classical set, any temperature can only be categorized into one subset, either LOW, MEDIUM or HIGH, and the boundary is crystal clear. But in the fuzzy set such as shown in Figure 2.3b, these boundaries becomes vague or smooth. One temperature can be categorized into two or maybe even three subsets simultaneously. For example, the temperature 40 °F can be considered to belong to LOW to a certain degree, say 0.5 degree, but at the same time it can belong to MEDIUM to about 0.7 degree. Another interesting thing is the temperature 50 °F, which can be considered to belong to LOW and HIGH to around 0.2 degree and belong to MEDIUM to almost 1 degree. The dash-line in Figure 2.3b represents the classical set boundary.

It is clear that a fuzzy set contains elements which have varying degrees of membership in the set, and this is contrasted with the classical or crisp sets because members of a classical set cannot be members unless their membership is full or complete in that set. A fuzzy set allows a member to have a partial degree of membership and this partial degree membership can be mapped into a function or a universe of membership values. Assume that we have a fuzzy set  $A$ , and if an element  $x$  is a member of this fuzzy set  $A$ , this mapping can be denoted as

$$\mu_A(x) \in [0, 1] \quad (A = (x, \mu_A(x)) | x \in X) \quad (2.10)$$

A fuzzy subset  $A$  with an element  $x$  has a membership function of  $\mu_A(x)$ . When the universe of discourse  $X$  is discrete and finite, this mapping can be expressed as

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots = \sum_i \frac{\mu_A(x_i)}{x_i} \quad (2.11)$$

When the universe  $X$  is continuous and infinite, the fuzzy set  $A$  can be represented as

$$A = \int \frac{\mu_A(x)}{x} \quad (2.12)$$

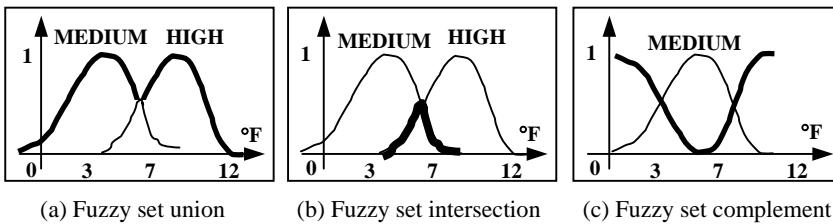


Figure 2.4. Fuzzy set operations

As we discussed in the previous section for the classical set operations, the basic fuzzy set operations also include intersection, union and complement, and those operations are defined as

$$\begin{array}{ll} \text{Union:} & A \cup B = \mu_A(x) \cup \mu_B(x) = \max(\mu_A(x), \mu_B(x)) \\ \text{Intersection:} & A \cap B = \mu_A(x) \cap \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \\ \text{Complement:} & A^c = F \setminus A \end{array} \quad (2.13)$$

where  $A$  and  $B$  are two fuzzy sets and  $x$  is an element in the universe of discourse,  $X$ . The graphical illustrations of those operations are shown in Figure 2.4.

According to the definition of fuzzy set operations, a fuzzy set union operation is exactly equivalent to select the maximum member from those members in the sets, and the intersection operation is selecting the minimum member from the sets.

### 2.2.4 Comparison Between Classical and Fuzzy Sets

Recall that in Section 2.2.1 we defined a college faculty set  $F$ , a faculty set for the computer science department, set  $A$ , and a set of faculty members who are under age 40, or set  $B$ . Now we can make a clear comparison for the classical and fuzzy sets using this example.

According to classical or crisp set theory, set  $B$  includes only three members who are under age 40 ( $x_1, x_2, x_3$ ), so the boundary between the faculty who are under age 40 and the faculty who are above age 40 is crystal clear (see solid-line in Figure 2.5). However, for the fuzzy set theory, set  $B$  contains not only those three members,  $x_1, x_2, x_3$ , but also some other members to some varying degree. The actual age distribution of all ten faculty members is displayed in the  $x$ -axis in Figure 2.5.

It can be found from Figure 2.5 that the member  $x_4$  and  $x_5$  do not belong to the set  $B$  in the view of the crisp set, but they can be considered as partial members in a partial membership of 0.7 and 0.3 degrees, respectively. Mapping this to a membership function, it can be expressed as

$$\mu_B(x) = \frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \frac{0.7}{x_4} + \frac{0.3}{x_5} \quad (2.14)$$

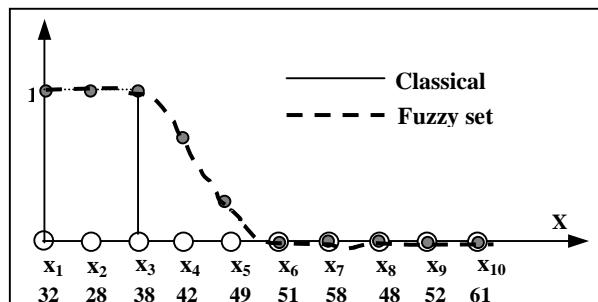


Figure 2.5. The comparison between classical and fuzzy sets

where the division operator is called a separator, and '+' is the or operator. All other members whose membership is 0 are omitted from this function.

### 2.3 Fuzzification and Membership Functions

The fuzzy set is a powerful tool and allows us to represent objects or members in a vague or ambiguous way. The fuzzy set also provides a way that is similar to a human being's concepts and thought process. However, just the fuzzy set itself cannot lead to any useful and practical products until the fuzzy inference process is applied. To implement fuzzy inference to a real product or to solve an actual

problem, as we discussed before, three consecutive steps are needed, which are: Fuzzification, fuzzy inference and defuzzification.

Fuzzification is the first step to apply a fuzzy inference system. Most variables existing in the real world are crisp or classical variables. One needs to convert those crisp variables (both input and output) to fuzzy variables, and then apply fuzzy inference to process those data to obtain the desired output. Finally, in most cases, those fuzzy outputs need to be converted back to crisp variables to complete the desired control objectives.

Generally, fuzzification involves two processes: derive the membership functions for input and output variables and represent them with linguistic variables. This process is equivalent to converting or mapping classical set to fuzzy set to varying degrees.

In practice, membership functions can have multiple different types, such as the triangular waveform, trapezoidal waveform, Gaussian waveform, bell-shaped waveform, sigmoidal waveform and S-curve waveform. The exact type depends on the actual applications. For those systems that need significant dynamic variation in a short period of time, a triangular or trapezoidal waveform should be utilized. For those system that need very high control accuracy, a Gaussian or S-curve waveform should be selected.

To illustrate the process of fuzzification, we still use the air conditioner example developed in the previous section. Assume that we have an air conditioner control system that is under the control of only a heater. If the temperature is high, the heater control motor should be turned off, and if the temperature is low, that heater motor should be turned on, which are common sense.

Regularly, the normal temperature range is from 20 °F to 90 °F. This range can be further divided into three sub-range or subset, which are

Low temperature:	20 °F ~ 40 °F, 30 °F is center
Medium temperature:	30 °F ~ 80 °F, 55 °F is center
High temperature:	60 °F ~ 90 °F, 75 °F is center

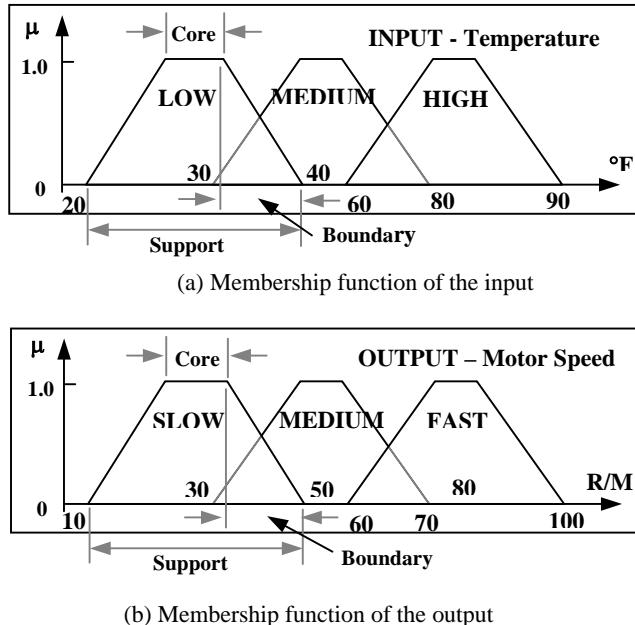
Next those three ranges need to be converted to linguistic variables: LOW, MEDIUM and HIGH, which correspond to the three temperature ranges listed above.

The membership function of these temperatures is shown in Figure 2.6. To make thing simple, a trapezoidal waveform is utilized for this type of membership function. A crisp low temperature can be considered as a medium temperature to some degree in this fuzzy membership function representation. For instance, 35 °F will belong to LOW and MEDIUM to 0.5 degree. Some terminologies used for the membership function are also shown in Figure 2.6.

The support of a fuzzy set, says LOW, is the set of elements whose degree of membership in LOW is greater than 0. This support can be expressed in a function form as

$$\text{Support}(\text{LOW}) = \{ x \in T \mid \mu_{\text{LOW}}(x) > 0 \} \quad (2.15)$$

It can be shown that the support of a fuzzy set is a classical set.



**Figure 2.6.** Membership function of input and output

The core of a fuzzy set is the set of elements whose degree of membership in that set is equal to 1, which is equivalent to a crisp set. The boundary of a fuzzy set indicates the range in which all elements whose degree of membership in that set is between 0 and 1 (0 and 1 are excluded).

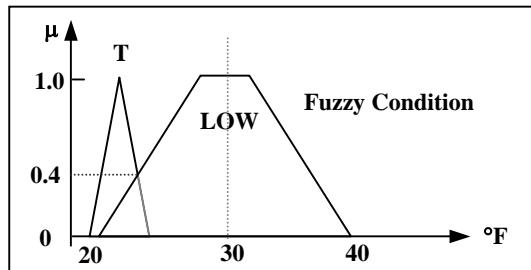
After the membership functions are defined for both input and output, the next step is to define the fuzzy control rule.

## 2.4 Fuzzy Control Rules

Fuzzy control rule can be considered as the knowledge of an expert in any related field of application. The fuzzy rule is represented by a sequence of the form IF-THEN, leading to algorithms describing what action or output should be taken in terms of the currently observed information, which includes both input and feedback if a closed-loop control system is applied. The law to design or build a set of fuzzy rules is based on a human being's knowledge or experience, which is dependent on each different actual application.

A fuzzy IF-THEN rule associates a condition described using linguistic variables and fuzzy sets to an output or a conclusion. The IF part is mainly used to capture knowledge by using the elastic conditions, and the THEN part can be utilized to give the conclusion or output in linguistic variable form. This IF-THEN rule is widely used by the fuzzy inference system to compute the degree to which the input data matches the condition of a rule. Figure 2.7 illustrates a way to

calculate the degree between a fuzzy input  $T$  (temperature) and a fuzzy condition LOW. Here we still use the air conditioner system as an example.



**Figure 2.7.** Matching a fuzzy input with a fuzzy condition

This calculation can also be represented by the function

$$M(T, \text{LOW}) = \text{Support} \min(\mu_T(x), \mu_{\text{LOW}}(x)) \quad (2.16)$$

Two types of fuzzy control rules are widely utilized for most real applications. One is fuzzy mapping rules and the other is called fuzzy implication rules.

### 2.4.1 Fuzzy Mapping Rules

Fuzzy mapping rules provide a functional mapping between the input and the output using linguistic variables. The foundation of a fuzzy mapping rule is a fuzzy graph, which describes the relationship between the fuzzy input and the fuzzy output. Sometimes, in real applications, it is very hard to derive a certain relationship between the input and the output, or the relationship between those inputs and outputs are very complicated even when that relationship is developed. Fuzzy mapping rules are a good solution for those situations.

Fuzzy mapping rules work in a similar way to human intuition or insight, and each fuzzy mapping rule only approximates a limited number of elements of the function, so the entire function should be approximated by a set of fuzzy mapping rules. Still using our air conditioner system as an example, a fuzzy mapping rule can be derived as

IF the temperature is LOW, THEN the heater motor should be rotated FAST.

For other input temperatures, different rules should be developed.

For most actual applications, the input variables are commonly more than one dimension. For example, in our air conditioner system, the inputs include both current temperature and the change rate of the temperature. The fuzzy control rules should also be extended to allow multiple inputs to be considered to derive the output. Table 2.1 is an example of fuzzy control rules applied in our air conditioner system.

**Table 2.1.** An example of fuzzy rules

$\dot{T}$	LOW	MEDIUM	HIGH
LOW	FAST	MEDIUM	MEDIUM
MEDIUM	FAST	SLOW	SLOW
HIGH	MEDIUM	SLOW	SLOW

The rows and columns represent two inputs, the temperature input and the change rate of the temperature input, and those inputs are related to IF parts in IF-THEN rules. The conclusion or control output can be considered as a third dimensional variable that is located at the cross point of each row (temperature) and each column (change rate of the temperature), and that conclusion is associated with the THEN part in IF-THEN rules. For example, when the current temperature is LOW, and the current change rate of the temperature is also LOW, the heater motor's speed should be FAST to increase the temperature as soon as possible. This can be represented by the IF-THEN rule as

IF the temperature is LOW, and the change rate of the temperature is LOW,  
THEN the conclusion or output (heater motor speed) should be FAST

All other rules follow a similar strategy, which is very similar to a human being's intuition.

In this air conditioner example, a total of nine rules are developed. For those applications that need high control accuracy, the input and output should be divided into more small segments, and more fuzzy rules should be applied.

#### 2.4.2 Fuzzy Implication Rules

A fuzzy implication rule describes a generalized logic implication relationship between inputs and outputs. The foundation of a fuzzy implication rule is the narrow sense of fuzzy logic [8]. Fuzzy implication rules are related to classical two-valued logic and multiple valued logic.

Still using the air conditioner system as an example, the implication is

IF the temperature is LOW, THEN the heater motor should be FAST.

Based on this implication and a fact: the temperature is HIGH. The result that the heater motor should slow down or the SLOW can be inferred.

#### 2.5 Defuzzification and the Lookup Table

The conclusion or control output derived from the combination of input, output membership functions and fuzzy rules is still a vague or fuzzy element, and this process is called fuzzy inference. To make that conclusion or fuzzy output

available to real applications, a defuzzification process is needed. The defuzzification process is meant to convert the fuzzy output back to the crisp or classical output to the control objective. Remember, the fuzzy conclusion or output is still a linguistic variable, and this linguistic variable needs to be converted to the crisp variable via the defuzzification process. Three defuzzification techniques are commonly used, which are: Mean of Maximum method, Center of Gravity method and the Height method.

### 2.5.1 Mean of Maximum (MOM) Method

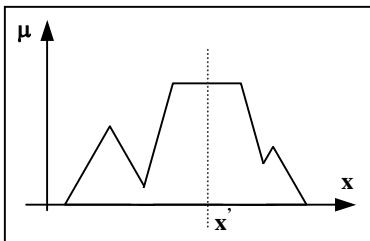
The Mean of Maximum (MOM) defuzzification method computes the average of those fuzzy conclusions or outputs that have the highest degrees. For example, the fuzzy conclusion is: the heater motor  $x$  is rotated FAST. By using the MOM method, this defuzzification can be expressed as

$$\text{MOM}(\text{FAST}) = \frac{\sum x}{T} \quad T = \{x \mid \mu_{\text{FAST}}(x) = \text{Support } \mu_{\text{FAST}}(x)\} \quad (2.17)$$

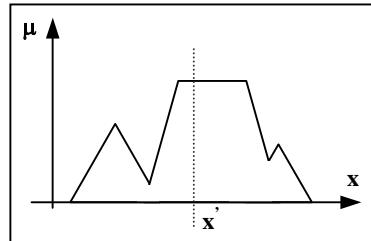
where  $T$  is the set of output  $x$  that has the highest degrees in the set FAST.

A graphic representation of the MOM method is shown in Figure 2.8a.

A shortcoming of the MOM method is that it does not consider the entire shape of the output membership function, and it only takes care of the points that have the highest degrees in that function. For those membership functions that have different shapes but the same highest degrees, this method will produce the same result.



(a) MOM method example



(b) COG method example

**Figure 2.8.** Graphic representation of defuzzification techniques

### 2.5.2 Center of Gravity (COG) Method

The Center of Gravity method (COG) is the most popular defuzzification technique and is widely utilized in actual applications. This method is similar to the formula

for calculating the center of gravity in physics. The weighted average of the membership function or the center of the gravity of the area bounded by the membership function curve is computed to be the most crisp value of the fuzzy quantity. For example, for the conclusion: the heater motor  $x$  is rotated FAST. The COG output can be represented as

$$\text{COG(FAST)} = \frac{\sum_x \mu_{\text{FAST}}(x) \times x}{\sum_x \mu_{\text{FAST}}(x)} \quad (2.18)$$

If  $x$  is a continuous variable, this defuzzification result is

$$\text{COG(FAST)} = \frac{\int \mu_{\text{FAST}}(x) x dx}{\int \mu_{\text{FAST}}(x) dx} \quad (2.19)$$

A graphic representation of the COG method is shown in Figure 2.8b.

### 2.5.3 The Height Method (HM)

This defuzzification method is valid only for the case where the output membership function is an aggregated union result of symmetrical functions [9]. This method can be divided into two steps. First, the consequent membership function  $F_i$  can be converted into a crisp consequent  $x = f_i$  where  $f_i$  is the center of gravity of  $F_i$ . Then the COG method is applied to the rules with crisp consequents, which can be expressed as

$$x = \frac{\sum_{i=1}^M w_i f_i}{\sum_{i=1}^M w_i} \quad (2.20)$$

where  $w_i$  is the degree to which the  $i$ th rule matches the input data. The advantage of this method is its simplicity. Therefore many neuro-fuzzy models use this defuzzification method to reduce the complexity of calculations.

### 2.5.4 The Lookup Table

The terminal product of defuzzification is the lookup table. Defuzzification needs to be performed for each subset of a membership function, both inputs and outputs. For instance, in the air conditioner system, one needs to perform defuzzification for each subset of temperature input such as LOW, MEDIUM and HIGH based on the associated fuzzy rules. The defuzzification result for each subset needs to be stored

in the associated location in the lookup table according to the current temperature and temperature change rate. In the following we use the air conditioner system as an example to illustrate the defuzzification process and the creation of the lookup table.

To make this illustration simple, we make two assumptions: I. assume that the membership function of the change rate of the temperature can be described as in Figure 2.9; II. only four rules are applied to this air conditioner system, which are

- 1) IF the temperature is LOW, and the change rate of the temperature is LOW, THEN the heater motor speed should be FAST
- 2) IF the temperature is MEDIUM, and the change rate of the temperature is MEDIUM, THEN the heater motor speed should be SLOW
- 3) IF the temperature is LOW, and the change rate of the temperature is MEDIUM, THEN the heater motor speed should be FAST
- 4) IF the temperature is MEDIUM, and the change rate of the temperature is LOW, THEN the heater motor speed should be MEDIUM

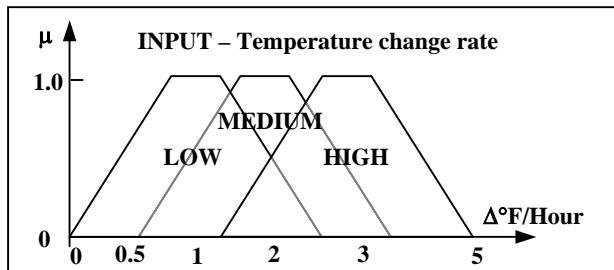
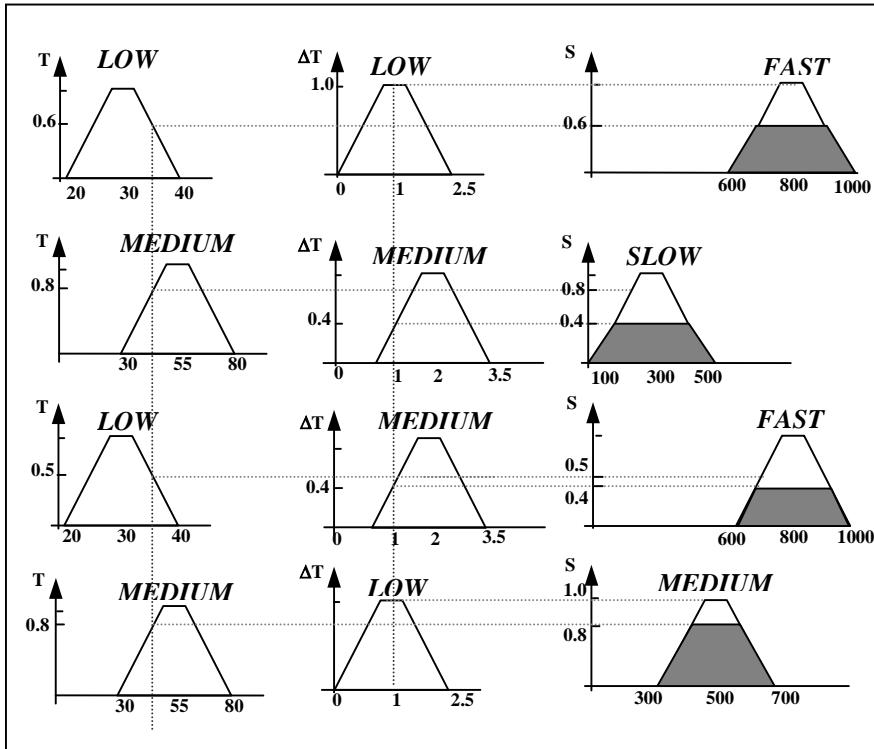


Figure 2.9. The membership function of the change rate of temperature

Based on the assumption made for the membership function and fuzzy rules, we can illustrate this defuzzification process using a graph. Four fuzzy rules can be interpreted as functional diagrams, as shown in Figure 2.10.

As an example, consider the current input temperature is 35 °F and the change rate of the temperature is 1 °F per hour.

From Figure 2.10, it can be found that the points of intersection between the temperature values of 35 °F and the graph in the first column (temperature input  $T$ ) have the membership functions of 0.6, 0.8, 0.5 and 0.8. Likewise, the second column (temperature change rate  $\Delta T$ ) shows that a temperature change rate of 1 °F per hour has the membership functions of 1.0, 0.4, 0.4 and 1.0. The fuzzy output for the four rules is the intersection of the paired values obtained from the graph, or the AND result between the temperature input and the temperature change rate input. According to Equation (2.13), this operation result should be:  $\min(0.6, 1.0)$ ,  $\min(0.8, 0.4)$ ,  $\min(0.5, 0.4)$  and  $\min(0.8, 1.0)$ , which produces to 0.6, 0.4, 0.4 and 0.8, respectively.



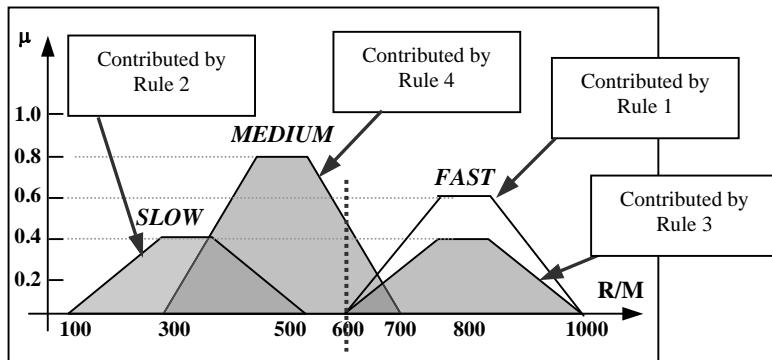
**Figure 2.10.** An illustration of fuzzy output calculation

The membership functions representing the control adjustment are weighted according to the input change and the different control contributions as shown in Figure 2.11.

Now, for a pair of temperature and temperature change rate, four sets of fuzzy outputs exist. To determine the crisp value of action to be taken from these contributions, one can either choose the maximum value using the MOM method or use the Center of Gravity method (COG). In this example, the COG method is used and the action is given by the center of the summed area, which is contributed by the different fuzzy outputs. Furthermore, the COG method gives a more reliable lookup table compared with the MOM operation.

Thus, for a temperature of 35 °F and a change rate of temperature is 1 °F per hour, the fuzzy output element  $y$  for this input pair is

$$y = \frac{0.6 \times 800 + 0.4 \times 300 + 0.4 \times 800 + 0.8 \times 500}{0.6 + 0.4 + 0.4 + 0.8} = 600 \text{ R/M} \quad (2.21)$$



**Figure 2.11.** Determination of fuzzy output by the center of gravity method

This defuzzified fuzzy output is a crisp or classical value, and should be entered into a certain location in a table called the lookup table. Since this fuzzy output element is associated with a temperature input pair with a temperature of 35 °F (belonging to LOW in the temperature membership function) and a change rate of temperature of 1 °F/h (also belonging to LOW in the membership function of change rate of temperature), this output value should be located in the cross point

between the LOW temperature  $T$  (row) and the LOW change rate  $\dot{T}$  (column) as shown in Table 2.2.

Table 2.2 shows an example of a lookup table.

**Table 2.2.** An example of a lookup table

$\dot{T} \backslash T$	LOW	MEDIUM	HIGH
LOW	600	?	?
MEDIUM	?	?	?
HIGH	?	?	?

To fill this table, one needs to use the defuzzification technique to calculate all other fuzzy output values and locate them in the associated positions in the lookup table as we did in this example. Generally, the dimensions of the fuzzy rules should be identical with the dimensions of the lookup table as shown in this example. To obtain more accurate control accuracy or fine fuzzy output element values, one can divide the inputs, say the temperature input and change rate of the temperature, into multiple smaller subsets to get finer membership functions. For instance, in this example, one can define a VERY LOW subset that covers 20 ~ 30 °F for the temperature input, a LOW subset that contains a range of 30 ~ 40 °F, and so on. Performing the same process to the change rate of temperature and the heater motor output, one can get a much finer lookup table.

For higher control accuracy applications, an interpolation process can be added after the lookup table to obtain finer output [10].

When implementing a fuzzy logic technique to a real system, the lookup table can be stored in a computer's memory, and the fuzzy output can be obtained based on the current inputs. In practice, there are two ways to calculate the lookup table using fuzzy inference process in the real control applications: off-line and on-line methods.

### 2.5.5 Off-line and On-line Defuzzification

As mentioned, the defuzzification process is to derive the desired crisp output value using a defuzzification technique that combines the membership functions with the fuzzy rules. The defuzzification process can be further divided into two categories: off-line defuzzification and on-line defuzzification.

So-called off-line defuzzification means that all input and output membership functions, fuzzy rules and the lookup table should be developed based on the estimations of the real application prior to the implementation of the fuzzy logic technique to the actual application. This means that all input and output membership functions are developed based on the actual experience or the input and output parameter ranges of a specified application, and the lookup table is calculated in terms of those definitions of input and output membership functions. The advantage of this method is that most fuzzy inference related calculations are performed prior to the real implementation and therefore the fuzzy process is less time consuming. The disadvantage of this technique is that the fuzzy output is only based on the estimation of input and output parameters, so the control accuracy is not as high as that of on-line method.

The on-line method has real-time controllability. Both input and output membership functions are developed during the real-time processing of an actual application. Also the lookup table elements are calculated in real-time based on the current actual inputs and outputs. In this method, only fuzzy rules are developed prior to the real applications. The advantage of this method is that higher control accuracy can be obtained for a process and the fuzzy output can be computed in real-time. The disadvantage of this method is that a longer processing time is needed and it is a somewhat time-consuming process. However, with the development of new computer technologies, today much faster CPUs are available, and processing time is no longer a big deal for this method.

## 2.6 Architectures of Fuzzy Logic Controls

Combining the discussions we made in the previous sections, a structure or architecture of fuzzy logic control system is given here.

As shown in Figure 2.12, which is a typical fuzzy closed-loop control system, the inputs are error and error rate, which are combined by block  $M$  to input to the fuzzy inference system. The lookup table is derived based on the membership function of inputs, the output and the fuzzy control rules. A control gain factor  $G$  is used to tune the output of the lookup table to obtain different output values. The

interpolation block S is used to smooth the output element of the lookup table. A feedback signal is obtained from the output of the system.

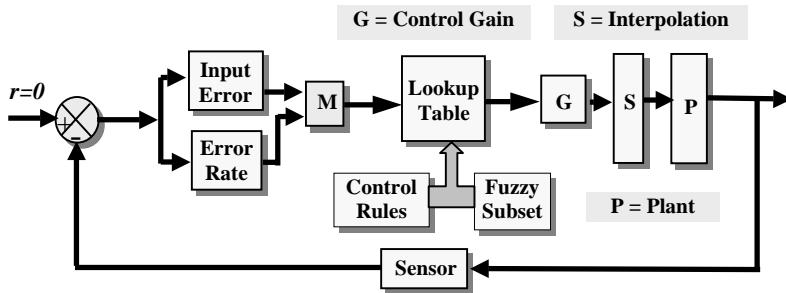


Figure 2.12. Block diagram of a fuzzy control system

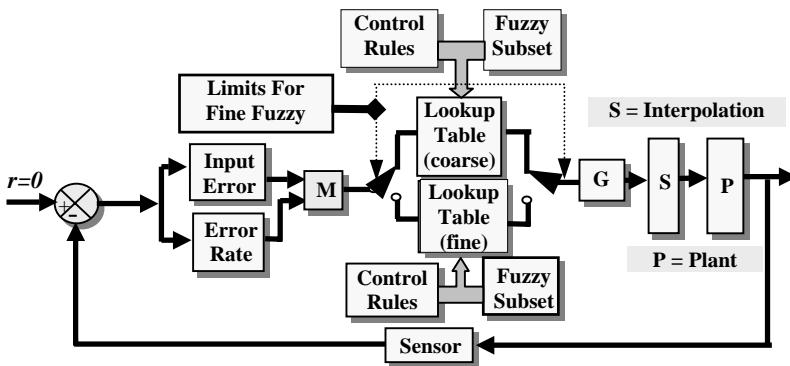


Figure 2.13. Block diagram of an accurate fuzzy control system

For a system that needs higher control accuracy, a multiple lookup tables fuzzy control system is needed, which is shown in Figure 2.13.

Two lookup tables are developed in this control system, a coarse and a fine table. During the application, the switch between the coarse and the fine table is under the control of the input error limit. This limit value can be defined by the user based on the real application. Two-set membership functions and control rules are utilized in this system to satisfy the requirement of higher control accuracy. When the system needs quick responses or quick actions, the coarse table is used. When the system needs high control accuracy or small control errors, the fine lookup table is selected. The sacrifice for this method is that more memory is needed to store both coarse and fine tables, and a little longer time is needed to make decision in selecting table in terms of the input error limit value.

## 2.7 Summary

A review of the fundamentals of fuzzy sets, fuzzy rules and fuzzy inference systems is provided in this chapter. Beginning with crisp or classical sets and their operations, we derived fuzzy sets and their operations. Classical set membership functions and fuzzy membership functions are discussed in detail following set theory. Fuzzy rules are described using an air conditioner control example. The different defuzzification techniques and their processes are discussed with the same example step by step. Finally, some other fuzzy techniques are discussed such as off-line and on-line fuzzy control systems as well as a fuzzy closed-loop control system including multiple lookup tables.

Because of space limitations, the author cannot cover all different fuzzy systems and their applications such as sliding mode fuzzy, adaptor fuzzy and neural fuzzy control systems in one chapter. With the development of new fuzzy techniques, fuzzy control will play a more and more important role in our society.

## References

- [1] Zadeh L. A. (1965) Fuzzy Sets. *Intl J. Information Control* 8:338-353.
- [2] Mamdani E. H., Assilion S (1974) An Experiment in Linguistic Synthesis With a Fuzzy Logic Controller, *Intl J. Man-Machine Stud* 7:1-13.
- [3] Holmblad L. P., Ostergaard J. J. (1982) Control of Cement Kiln by Fuzzy Logic, Gupta M. M, Sarchez E, Fuzzy Information and Decision Processes, North Holland, pp. 389-399.
- [4] Meng Joo Er and Ya Lei Sun, "Hybrid Fuzzy Proportional Integral Plus Conventional Derivative Control of Linear and Nonlinear Systems", *IEEE Trans. On Industrial Electronics*, Vol.48, No.6 December 2001, pp. 1109-1117.
- [5] J. C. Wu and T. S. Liu, "A Sliding-Mode Approach to Fuzzy Control Design", *IEEE Trans. On Control Systems Technology*, Vol.4, No.2, March 1996, pp. 141-151.
- [6] Feng-Yi Hsu and Li-Chen Fu, "Intelligent Robot Deburring Using Adaptive Fuzzy Hybrid Position/Force Control", *IEEE Trans. On Robots and Automation*, Vol.16, No.4, August 2000, pp.325-334.
- [7] H. X. Li and H. B. Gatlant, "A New Methodology for Designing a Fuzzy Logic Controller", *IEEE Trans. On Sys., Man, and Cybernetics*, Vol.25, No.3, March 1995, pp. 505-512.
- [8] John Yen and Reza Langari (1999) Fuzzy Logic – Intelligence, Control, and Information, Prentice Hall.
- [9] Mohammad Jamshidi, Nader Vadiee and Timothy J. Ross (1993), Fuzzy Logic and Control, Prentice Hall.
- [10] Ying Bai, Hanqi Zhuang and Zvi. S Roth, "Fuzzy Logic Control to Suppress Noises and Coupling Effects in a Laser Tracking System", *IEEE Trans on Control Systems Technology*, Vol.13, No.1, January 2005, pp.113-121.

## Implementation of Fuzzy Logic Control Systems

Dali Wang and Ying Bai

### 3.1 Overview

A common feature of conventional control is that the control algorithm is analytically described by equations. In general, the synthesis of such control algorithms requires a formalized analytical description of the controlled system by a mathematical model [1]. The main characteristic of fuzzy logic is its capability of expressing knowledge in a linguistic way, allowing a system to be described by simple, human friendly rules [2,3]. This alternative linguistic approach leads to a meaningful reduction of the computational burden. Usually fuzzy logic solutions are not aimed at achieving the computational precision of traditional techniques, but aim at finding acceptable solutions in a shorter time. The natural applications of fuzzy logic control (FLC) are those systems whose mathematical model is unknown, very complex or time-varying, and where human experience can play an important role.

As in other control applications, fuzzy logic implementation could be carried out either with general-purpose systems or dedicated systems, and many solutions have been proposed for both alternatives. The structure and relative simplicity of fuzzy processing algorithms naturally lead to straightforward implementation in dedicated hardware structures. However, the majority of fuzzy logic implementations reported in the literatures use general-purpose hardware due to the fact that such an implementation involves a low start-up cost with a well-defined control algorithm such as FLC. An optimal solution for the full application range of fuzzy logic does not exist. Different approaches have to be chosen based on the features of the application: complexity, real-time constraints, development time, quantities, and other factors that influence the choice of a design.

In general, we can classify four classes of FLC implementation alternatives:

- 1) Software solutions with general-purpose microprocessors.
- 2) Software solutions with general-purpose microprocessors with instructions for fuzzy logic specialized computation.
- 3) Hardware solutions using dedicated fuzzy processors.
- 4) Hardware solutions using fuzzy ASICs.

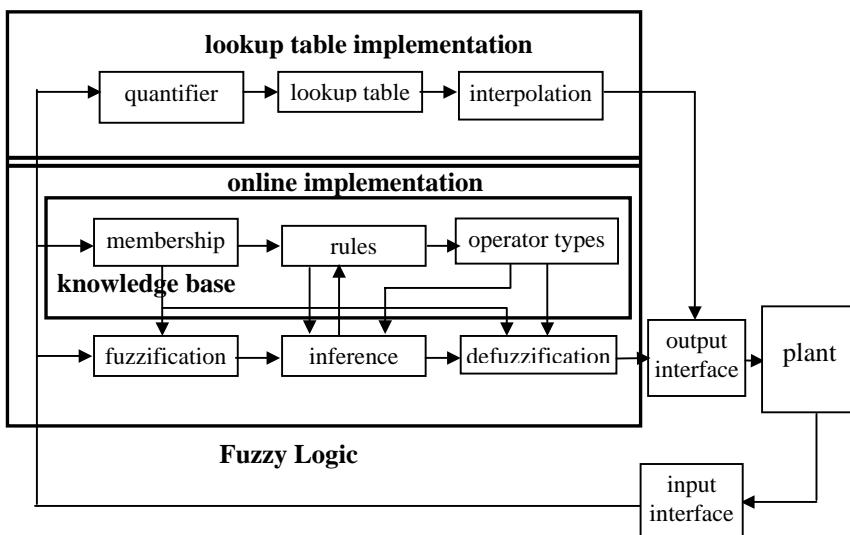
Other than technical merits, the stages of development and prototyping, as well as production volume, also contribute to the selection of different solutions for certain performance and cost criteria. For example, if the features of an application

are fixed and the volume is high, an ASIC solution may be an ultimate choice. However, for the same application, if the development time is very limited, a choice of microprocessor could be appropriate for initial stage development.

In this chapter, a number of implementation options for fuzzy logic control applications are presented. We start from an analysis of the computational load of all the processing stages of an FLC application. This lays a foundation for the following sections. The next section discusses software implementation techniques using general-purpose microprocessors, or general-purpose microprocessors with fuzzy logic specific instructions. It is followed by a section on hardware processing techniques using either dedicated fuzzy processors, or fuzzy ASICs. The conclusions are in the last section.

### 3.2 Computational Complexity of Fuzzy Logic Control Algorithms

The fundamental design of a fuzzy controller is to provide an input-output relationships in term of fuzzy rules. FLC makes in real-time three essential processing operations: fuzzification, fuzzy inference and defuzzification [1-4]. The focus of this section is not only a brief overview of the three processing steps, but also the computational load as well as the memory requirement involved in these operations. Figure 3.1 provides an overview of the various elements and operations involved in an FLC.



**Figure 3.1.** A fuzzy logic controller in several configurations

### 3.2.1 Fuzzification

Fuzzification is the operation that translates a crisp data input to a membership degree (also called  $\alpha$  level). The membership degree depends on the shape of membership function used, which has been extensively discussed in the literature [1][3-7]. There are two commonly-used methods to evaluate membership functions.

#### *A. Memory oriented approach*

This approach applies to the cases where the membership functions do not change at run time, and the total numbers of input and membership functions are limited. The alpha values for each input and output are evaluated off-line and stored in memory as discrete values. An easy way to choose the array size (dimension) for the membership functions is to set the array size equal to the resolution of the analog-to-digital converter (ADC). For example, with an 8-bit ADC input, arrays of size 256 could be used to represent input membership functions [8]. Programming the membership function is a simple matter of converting the membership functions to data arrays and storing them in memory [9].

In this case the shape of the membership function does not influence the computational load of the algorithm. However, the memory usage depends on a number of factors: the number of inputs, the resolution of input value and membership function value, and the number of membership functions. The total memory requirement for input membership functions is given by the following equation:

$$MEM = N_x \cdot N_{mf-x} \cdot 2^{BITS-X} \cdot BYTE_{mf} \quad (3.1)$$

where  $N_x$  is the number of inputs,  $N_{mf-x}$  is the number of membership functions for each input,  $BITS-X$  is the number of bits used by each input value (i.e., the resolution of analog-to-digital converter or ADC), and  $BYTE_{mf}$  is the number of bytes used to represent a membership function degree. We assume that all inputs use the same number of bits, and that all membership functions use the same number of bytes to represent a degree; otherwise, the total memory would need to be calculated on each individual input, which is followed by a summarization operation.

For example, a fuzzy controller contains three input values that are obtained through a 10-bit ADC, each input is defined upon five linguistic labels: large positive, small positive, zero, small negative, large negative, and one byte is used to store a membership degree. The memory requirement for the input membership functions is 15K bytes.

#### *B. Computation oriented approaches*

The evaluation of the membership degree is made on-line. In this case, we only need to store the parameters of membership functions. Consequently, the shape of the membership function is a very important aspect since the burden on computation could be significant. The linear type of membership functions such as triangular and trapezoid shape are preferred choices since they involve less

computation. In the case of triangular membership functions, the necessary parameters are the gradient and the intercept of two straight lines composing the triangle. Execution time depends on the shape of the membership function. For triangular functions, we need two memory accesses, three comparisons, one addition and one multiplication to compute a single alpha value.

To summarize, we present the following guidelines for membership function implementation:

- 1) The memory-oriented approaches involve only memory access operation and no computational effort, thus providing fast access to alpha values. We can also get membership functions of any shape with this method. However, it only applies to systems with limited numbers of inputs, and limited types of membership functions. In addition, the membership functions have to be static – no run-time changes are possible.
- 2) The computation-oriented approaches use less memory space at the expense of computational load. Unless there is a good reason to do otherwise, use a triangular or trapezoidal membership function.

### *C. Final comments*

The discussion above, though targeting input membership functions, applies equally well to output membership functions. All the techniques, equations and principles apply to the implementation of both input and output membership functions.

There is no clear distinction between the memory-oriented and the computation-oriented approaches. As a point of fact, many systems are implemented using the combination of two techniques. A common case is that a system has many inputs but only one output. In this case, a computation-oriented approach could be used for the fuzzification of inputs, where a memory-oriented approach (lookup table) could be used for the defuzzification process [10].

## **3.2.2 Fuzzy Inference**

Fuzzy Inference is the operation that uses the rule base and the alpha values to deduce the fuzzy output. This step can be accomplished in many different ways. The most classical inference models in FLC are the Mamdani model and the TSK model [1, 3, 4, 6].

### *A. The Mamdani model:*

The Mamdani model usually employs either the max-min method or the max-dot method [1, 3, 4, 6]. Both methods generate an inferred conclusion by suppressing the membership function of the consequence.

The fuzzy inference involves two steps: finding the output of each of rule, and aggregating the individual rule outputs to obtain the overall system output. Suppose that there are  $R$  rules and the  $i$ th. rule has the form

IF  $x_1$  is  $A_{i1}$ , and  $x_2$  is  $A_{i2}$ , ..., and  $x_N$  is  $A_{iN}$ , THEN  $y$  is  $B_i$

The output of the  $i$ th. rule is first to combine the matching levels (alpha level) of all fuzzy variables using *anding*-type aggregation:

$$\alpha_i = \wedge(A_{i1}(x_1), A_{i2}(x_2), \dots, A_{iN}(x_N)) \quad (3.2)$$

The result interacts with its consequence  $B_i$  to provide the output of the rule. In FLC the most commonly used method for inferring the rule output is the so-called Mamdani method. In the Mamdani method, the output fuzzy set  $R_i$  is obtained by *adding* of the above obtained matching level:

$$\mu_i(y) = \alpha_i \rightarrow B_i(y) \quad (3.3)$$

The next step is the aggregation of the individual rule outputs to obtain the overall system output  $\mu(y)$ , a fuzzy set of output  $y$ . The individual rule outputs are aggregated using an *oring* aggregation:

$$\mu(y) = \vee_i \mu_i(y) \quad (3.4)$$

The algorithm of FLC is based on three paradigms: the AND operation between the antecedents of the rules; the OR operator between the individual rules; and the fuzzy implication interpreted as an ADD operation. A common set of interpretations for the three operators is: min for AND, max for OR, and min for ADD. This becomes the min-max method. Another commonly used interpretation of these operator is to use multiplicative operation for ADD and/or AND operator. This results in the max-dot method.

For the min-max method, if each rule involves all  $N_x$  input variables (worst case) and we have  $R$  rules, we must execute  $R(N_x - 1)$  cascaded minimum operations. For the max-dot method where min is used for AND and multiplication is used for ADD, there are extra multiplication operations involved in addition to the minimum operation. The maximum number of multiplication operation for each output is  $R \cdot 2^{\text{BITS-}Y}$ , where BITS-Y is the number of bits used by the output.

Notice that the output  $\mu(y)$  is a fuzzy set. The fuzzy output can be defuzzified into a crisp output using one of the defuzzification techniques discussed in Chapter 2 and outlined in Section 3.2.3

### B. The TSK model

In general, rules in a TSK model have the form

IF  $x_1$  is  $A_{i1}$ , and  $x_2$  is  $A_{i2}$ , ..., and  $x_N$  is  $A_{iN}$   
THEN  $y = f_i(x_1, x_2, \dots, x_N)$

where

$$f_i(x_1, x_2, \dots, x_N) = b_{i0} + b_{i1}x_1 + \dots + b_{iN}x_N \quad (3.5)$$

The degree the  $N$  inputs match the  $i$ th rule is computed using the *anding*-type operator (min in this case) as Equation (3.2). The output of  $i$ th rule is computed using Equation (3.5).

If each rule involves all  $N_x$  input variables (worst case) and we have  $R$  rules, we must execute  $R(N_x - 1)$  cascaded minimum (comparison) operations, followed by  $N_x$  multiplication and summation operation.

The total output of the model is given by

$$y = \frac{\sum_i \alpha_i f_i(x_1, x_2, \dots, x_N)}{\sum_i \alpha_i} \quad (3.6)$$

Notice that the output  $y$  is a crisp value.

### 3.2.3 Defuzzification

This is the step that translates the fuzzy output to a crisp output. Many defuzzification methods have been proposed in the literature [1][3-7]. The most commonly used are the center of gravity (COG) method, the mean of maximum method (MOM) and the height method (HM).

The COG method defines the defuzzified value of a fuzzy set as its fuzzy centroid:

$$y = \frac{\sum_j \mu(y_j) y_j}{\sum_j \mu(y_j)} \quad (3.7)$$

The MOM method determine the defuzzified value as a mean of all values of the universe of discourse, having maximum membership grades:

$$y = \frac{\sum_{y_j \in T} y_j}{|T|} \quad (3.8)$$

where  $T$  is the set of elements which attain the maximum value of  $\mu(y)$ , and  $|T|$  is the cardinality of  $T$ .

The HM is based on the center of gravity of each rule, weighted by the matching levels of each rule:

$$y = \frac{\sum_i \mu_i f_i}{\sum_i \mu_i} \quad (3.9)$$

where  $f_i$  is the center of gravity of each output membership function.

The following worst-case analysis applies to each output with  $2^{\text{BITS-}Y}$  discrete levels,  $N_{\text{mf-}y}$  as the numbers of output membership functions. The COG method also needs  $2^{\text{BITS-}Y}$  multiplication-and-accumulation operations,  $2^{\text{BITS-}Y}$  summation operations, and one division operation. The HM needs  $N_{\text{mf-}y}$  multiplication-and-accumulation operations,  $N_{\text{mf-}y}$  summation operations, and one division. The MOM needs  $2^{\text{BITS-}Y}$  summations, one comparison, and one division.

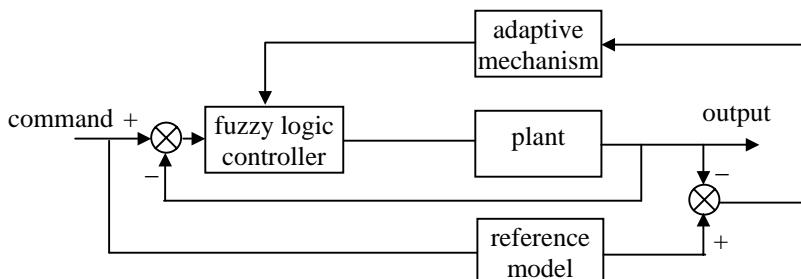
In general, the computation loads by the three methods in a descending order are COG, HM, and MOM. The COG is the most commonly used method for defuzzification.

### 3.2.4 Static or Dynamic Fuzzy Logic Control

The control algorithms could be either static or dynamic (adaptive). In the static case, the set of control policies does not change in time. In the dynamic case, the set of control policies changes over time, often in response to plant state and environmental conditions. This also applies to fuzzy logic controllers. In a static FLC case, designs are done off-line and implemented in real time. There is no real time adjustment of the algorithm based on the running environment, even though configuration changes are possible. In a dynamic case, the control algorithm could be adjusted adaptively in run time.

Adaptive control is a rich area of study, reflecting the difficulty in constructing controllers that tolerate uncertainty. It has been pointed out in [11] that adaptive control is concerned with two fundamental aspects of control in the face of incomplete plant characterization. The first is identification, in which we seek to determine the characteristics and state of the plant under control. The second aspect is the challenge of determining control. The issues of identification and control appear in an adaptive FLC as the two related problems: how much controller adjustment is needed, and which controller components should be adjusted. These include the adjustment of membership functions, rules, parameters used by the controller, *etc.* [5,12]. The adaptive FLC block diagram is shown in Figure 3.2, which illustrates the construction used by the majority of the applications.

The adaptive FLC is not the focus of this chapter. The topics of this chapter only cover the Fuzzy Logic Controller block in Figure 3.2. It could be a static fuzzy logic controller, or an adaptive one that is adjusted by the Adaptive Mechanism block.



**Figure 3.2.** Direct adaptive FLC block diagram

### 3.3 Software Solutions

In this section we will discuss the two commonly used platforms for software implementations of FLC: general-purpose microprocessors and general-purpose microprocessors with fuzzy support. We will then focus on the two types of implementation techniques: online and the lookup table approach.

#### 3.3.1 General-purpose Microprocessors

The term general-purpose microprocessors refers to a category of microprocessors called microcontrollers that are designed for industrial embedded applications. The microcontroller selected for FLC application should be optimized for demanding embedded applications where real-time control, digital signal processing (DSP) capability, and special peripheral functions are needed. The DSP capability is characterized by a built-in multiplier and a dedicated group of DSP instructions. The peripheral set typically includes a flexible timer unit, PWM generation, a fast multi-channel ADC, asynchronous and synchronous serial interfaces. Some of them also carry non-volatile memory for permanent program and data storage. There are many hardware manufacturers offering this category of products. Some of the manufacturers are Intel, Micro Chip, Texas Instrument, Motorola, STM and Infineon.

The advantages and disadvantages of the approach are shown as follow:

Pros:

- Relative short development time
- Complete flexibility
- Support of non-fuzzy computation by itself
- Availability of development systems and support tools

Cons:

- Low speed

#### 3.3.2 General-purpose Microprocessors with Fuzzy Support

An alternative approach is to maintain the general computational capability of a microcontroller while adding some fuzzy logic specialized operations to accelerate the application. The advantages and the disadvantages of this approach are shown below:

Pros:

- All the advantages of the general-purpose microprocessor
- Higher speed compared to standard microcontrollers
- Simple extension of existing cores and non-fuzzy computation

Cons:

- Limited hardware platform to choose from
- Possibly higher cost compared to standard microcontrollers (due to production volume)
- Limited performance compared to hardware solutions

Two examples in this category are given to offer a glimpse of some features of the products.

The Motorola HC12 microcontroller is a high-speed 16-bit evolution of 68HC11 architecture. This 16-bit microcontroller family includes four fuzzy logic instructions in addition to the memory and on-chip peripheral functions you would expect in a general-purpose microcontroller. The fuzzy logic instructions use existing CPU logic to perform computations including addition, subtraction, multiplication, multiply-and-accumulate, and comparisons. As a result, the speed and efficiency of fuzzy logic programs are greatly improved without increasing the cost of the microcontroller. All of the dedicated fuzzy logic instructions are designed to complement the need to a specific need in FLC. For example, the MEM instruction computes one fuzzy input based on a trapezoidal membership function; designed for fuzzy inference, the REV instruction performs unweighted min-max rule evaluation on a complete rule list for a fuzzy logic system; the WAV instruction computes a sum-of-products and a sum-of-weights needed for weighted average defuzzification.

ST Microelectronics also developed a family of microcontroller products, the ST FIVE family, that offers a traditional microcontroller architecture, characterized by an extended instruction set and a wide range of on-chip peripherals, combined with a dedicated architecture, the Decision Processor, that provides hardware instructions for fuzzy algorithms [13]. A software environment called Visual FIVE IDE is provided to develop fuzzy algorithms through a guided editor and a graphical tool for fuzzy sets design in the form of “if ... then” fuzzy rules. The developer could start from the high level FLC design using the development environment; the design is then translated into a set of instructions for the Decision Processor.

### 3.3.3 Software Development Tools for Fuzzy Logic System Design

There is a noticeable trend in the development of FLC algorithms. There are a number of software products available that would allow the user to design a fuzzy controller from a high or algorithmic level, and sometimes interactively with a special graphic user interface (GUI). The user could start from a fuzzy logic algorithmic level to design the controller. Consequently, these tools would usually generate C codes, which can be modified to fit into a target platform. This further simplifies the implementation of FLC using a general-purpose platform.

An interesting example is a platform-independent fuzzy logic development tool called *Fuzz-C*. It is a stand-alone preprocessor that integrates fuzzy logic into the C language. *Fuzz-C* accepts fuzzy logic rules, membership functions, and consequence functions, and produces standard C source code that can be compiled by most C compilers.

Some of these products are even integrated with a certain compiler of a microcontroller, such that a seamless integration is possible from a high-level FLC design to a low-level microcontroller software implementation. An interesting example is a tool called *fuzzyTECH*. A *fuzzyTECH* Precompiler Edition generates a complete fuzzy logic system as highly optimized C code for a solution that could be integrated with other software on different hardware platforms. Then a different

so-called Edition of *fuzzyTECH* is available to fit the needs of various hardware platforms. For example, the edition for the Motorola HC12 microcontroller could generate assembly code for HC12 to ensure maximum computational performance using as little memory resources as possible.

### 3.3.4 Online Implementation

As discussed in a previous section, three steps are involved in an FLC implementation: fuzzification, fuzzy inference and defuzzification. If part or all of the three steps are programmed into a microcontroller and run in real-time, the implementation is an online implementation, as shown in Figure 3.1. The advantages and the disadvantages of this approach are shown below:

Pros:

- Flexibility: the implementation could be reconfigured by a user or other applications in run-time
- Potential for high control performance: by its runtime adjustment capability, and by choosing the right design
- Suitable for both static and dynamic FLC

Cons:

- Computational load

The fuzzification process could be either a memory-oriented approach by a lookup table or a computation oriented approach by storing the parameters of the membership functions. It is a trade-off between the memory usage and computational load. If the numbers of inputs and outputs are very limited, and all the membership functions are static, the memory-oriented approach would be attractive. Otherwise, choose the computation-oriented approach.

For the Mamdani model, the fuzzy inference processing employs either the max-min method or the max-dot method. Both need to complete the operation defined by Equations (3.2) to (3.4) in real time. The computational load for both methods is provided in Section 3.2, where the max-dot method requires extra multiplication operations in addition to comparison operations.

For the TSK model, fuzzy inference completes the operations defined by Equations (3.2) and (3.5) in real time. Where there is no defuzzification process for the TSK model, it needs to complete the operation defined by Equation (3.6) to obtain a control output.

For the purpose of defuzzification, one of the operations defined by Equations (3.7) through (3.9) needs to be complete in real time: center of gravity (Equation 3.7), height method (Equation 3.9), and mean of maximum (Equation 3.8) [14]. A general opinion is that the center of gravity method is preferred over the height method, which is preferred over the mean of maximum method. However, the center of gravity method is more computational intensive than the height method, due to the fact that the number of output membership functions is typically much fewer than the possible values of the fuzzy set over which the defuzzification process is applied.

Since the defuzzification process involves multiplication and division operations, the DSP capability of a microcontroller could significantly improve the

run time performance. Therefore, such a consideration needs to be given when selecting a hardware platform for an FLC implementation.

### 3.3.5 Lookup Table Implementation

Although an FLC always involves three operations, the design could be completed off-line. Consequently, we could find the corresponding output values for every possible set of input values. Then a table could be used to establish the mapping between all input combinations and output values, which is shown in Figure 3.1 as the lookup table implementation.

The example in Chapter 2 is quoted here. In a fuzzy control system involving an air conditioner system, the inputs include both current temperature and the rate of change of temperature. The output is the speed of a motor. If all input and output are 8-bit values, a lookup table shown in Table 3.1 could be built as the result of a fuzzy logic controller design.

**Table 3.1.** A lookup table example

T'	0	1	2	...	254	255
T						
0						
1						
2	Output Values					
:						
254						
255						

The advantages and the disadvantages of this approach are shown below:

Pros:

- Speed: there is only access time involved
- Easy to program

Cons:

- Only work for system with limited inputs and outputs
- No runtime flexibility: nothing could be adjusted/configured in real time

A main limitation to the lookup table method is that it only applies to systems with a limited number of inputs and outputs. The memory requirement grows exponentially with the increase of the number of inputs. The memory requirement by this approach is given by the following equation:

$$MEM = N_y \cdot (2^{BITS-X})^{N_x} \cdot BYTE_y \quad (3.10)$$

where  $N_x$  is the number of inputs,  $N_y$  is the number of outputs,  $\text{BYTE}_y$  is the numbers of bytes that all output values take,  $\text{BITS}-X$  is the number of bits used by each input values (*i.e.*, the resolution of ADC). We assume that the bit length for all inputs is the same; the byte length for all outputs is also the same. For example, an FLC has three inputs and one output. The input variables are sampled through a 10-bit ADC, each output variable only taking one byte (8 bits DAC). The memory requirement for the lookup table is 1,048,576 KB. Present-day low-cost microcontrollers do not have such a large on-chip memory.

Table 3.2 represents the memory requirement (in KB) for a lookup table for each output variable of one byte in length.

**Table 3.2.** An illustration of memory requirement (KB) for the lookup table approach

Number of input	Number of bits of input			
	6	8	10	12
1	0.0625	0.25	1	4
2	4	64	1,024	16,384
3	256	16,384	1,048,576	67,108,864
4	16,384	4,194,304	1,073,741,824	2.74878E+11

An alternative to the full-scale lookup table approach is to reduce the resolution of the input variables, since it affects the memory size exponentially. For the above example, if the lookup table were built upon 8-bit resolution of input variables instead of 10 bits, the memory requirement would become 16,384 KB. However, an interpolation step is needed in order to find the output value for a set of input values. Commonly used techniques include Bilinear and Cubic Spline. This would reduce the accuracy of the output, and affect the performance of the controller. In addition, the interpolation step consumes extra computational power.

In general, a direct lookup table method only works well for a system with no more than two inputs, and the resolution of input is no more than eight bits.

## 3.4 Hardware Solutions

### 3.4.1 Dedicated Fuzzy Processors

Most of the dedicated fuzzy processors fall into the category of coprocessors. These are the special-purpose processors dedicated to fuzzy computation, although they are not capable of implementing the entire control systems due to the lack of general-purpose computation capabilities. The advantages and disadvantages of this approach are shown below:

Pros:

- High speed
- Modularity

- Some flexibility (in term of configurability)

Cons:

- Can not function standalone, has to work with a general-purpose processor
- Possibly high cost (due to limited production volume)
- Limited flexibility (rules and membership functions)

These dedicated fuzzy processors generally function as a co-processor to a general-purpose microprocessor acting as a host processor. The two processors communicate through a high-speed interface. The host processor handles the non-fuzzy logic part of the system. It is also commonly used as an interface to the fuzzy processor by sending commands and configuration parameters to it. The designs of the fuzzy processors usually adopt a triangular/trapezoidal membership function, flexible rule format (the number of antecedents and consequents), the max-min inference method, eight bits input/output precision, and the center of gravity defuzzification method.

The dedicated fuzzy processors can be used for high speed and moderately complex applications. The typical processing time ranges from a few  $\mu$ s to a few hundred  $\mu$ s, making them suitable for plants with frequency response in the range of 10~100 kHz.

Among many developed solutions, one by SGS-THOMSON (ST) is briefly introduced here as an example. SGS-THOMSON developed a family of fuzzy digital coprocessors called WARP (Weight Associative Rules Processor) [15]. It is rather fast and configurable, targeting medium to high-end applications. It has been reported that with a simple application (2 inputs, 1 output, 7 rules, 5 membership functions and 8-bit resolution), the reported processing time with the fuzzy processor is less than one  $\mu$ s.

**Table 3.3.** The features of a family of fuzzy processors

WARP 1.1	WARP 2
Up to 256 rules (4 antecedents, 1 consequent)	Up to 256 rules (4 antecedents, 1 consequent)
Up to 16 input variables	Up to 8 input variables
Up to 16 membership function for input	Up to 16 membership function for input
Membership function with any shape	Triangular and trapezoidal membership functions
Up to 16 output variable	Up to 4 output variable
Max-dot inference method	Max-dot inference method
Computation time: 14 $\mu$ s for 256 rules	Computation time: 200 $\mu$ s for 256 rules
Clock frequency 40 MHz	Clock frequency 40 MHz

The key features of the products are illustrated in the Table 3.3, where WARP 2 is a low cost implementation of WARP1.1.

### 3.4.2 Fuzzy ASIC

For low cost, high speed and small size, the ultimate solution for FLC applications is to develop dedicated hardware for a single application. The most commonly used fuzzy ASICs have appeared in washing machine controls, auto focus imaging, speed and brake control for cars, and so on. Due to the process involved in an FLC, it is possible to design directly a dedicated hardware for a fuzzy logic algorithm in a way that maximizes speed and minimizes silicon area, by using high-level synthesis techniques [2]. The resultant solution is then tailored to a single application, and configurable parameters are reduced to a minimum that does not compromise speed and area. The advantages and disadvantages of this approach are shown below:

Pros:

- Speed: very fast
- Low cost (for high production volume)
- Possible as a single chip solution for the whole control system

Cons:

- No flexibility (designed for a single application)
- Possibly long overall design turnaround time (including prototyping and testing)
- No other processing capability – may need a microprocessor for other processing

The natural choice of ASIC solution is in the areas of high speed, fixed functionalities, and high production volume. When the speed of a dedicated fuzzy coprocessor cannot meet the timing constraints of an application, a fuzzy ASIC solution should be considered.

Many ASIC implementations have been developed for FLC [12, 16-18]. The challenge is to utilize a flexible architecture that fits different applications to make automatic synthesis feasible. Research has been conducted on the design of architectures suitable for automatic synthesis of digital fuzzy controllers. One proposed architecture uses the number of bits for input and output and their membership functions as main parameters, while imposing no limitations on the number of rules comprising the knowledge base [16].

A fuzzy system is inherently parallel in its representation: separate input variables, rules, and elements into which the output space is generated. Regarding hardware, this means a trade-off between high inference speed (parallel processing) and low silicon area (sequential processing). Singleton fuzzy systems that employ singleton values to define the consequents are usually chosen for hardware realization since they eliminate parallelism in output space. In the literature, fully digital approaches have been reported, reducing parallelism in rule inference by only processing some simultaneously active rules [18].

Every ASIC implementation of each FLC is different. However, most of them contain similar blocks due to the same type of operations involved. In the

fuzzification stage, the membership function could be stored in memory space for easy access. The inputs received by the fuzzy rule base are used to address the membership memory. The addresses of the membership values are computed by adding the crisp input to the base address of each membership function. The next step is fuzzy inference. From a hardware point of view, this corresponds to the execution of a set of minimum operations for each rule. The final step, defuzzification, performs the computation of the crisp output from their fuzzy values, according to one of the defuzzification algorithms (COG, MOM or HM). This operation could be the most expensive and time consuming area as it may require a multiplication-and-accumulation operation.

The realizations are strictly based on the application requirements. For instance, the system can use different types of adder (carry select, look ahead, *etc.*) and multipliers (parallel, sequential) according to performance and area constraints.

### 3.5 Conclusions

This chapter is intended to provide an overview of various approaches of FLC implementation. There are different hardware/software tradeoffs that must be considered in the design of a fuzzy controller. The different techniques provide different degrees of flexibility as well as supporting different ranges of applications in terms of speed and complexity. While there is a never-ending debate regarding the merits of hardware *vs* software, every design has to look into its own constraints for a suitable solution. In one extreme, for low speed applications with only one or two low-resolution inputs and a few outputs, a lookup table scheme could meet the need. In the other extreme, for a very high speed application with relative large number of inputs and output, a fuzzy ASIC solution could be the only choice. A system designer has to make a proper choice according to the applications and the cost/performance requirements.

## References

- [1] R. Yager, D. Filev, *Essentials of Fuzzy Modeling and Control*, Wiley, 1994.
- [2] A. Costa, A. De Gloria, P. Faraboschi, A. Pagni, G. Rizzotto, “Hardware Solutions for Fuzzy Control”, Proceedings of the IEEE, Vol. 83, Issue 3, March 1995, pp 422-434.
- [3] J. M. Mendel, “Fuzzy Logic Systems for Engineering: A Tutorial”, Proceedings of the IEEE, Vol. 83, Issue 3, March 1995, pp 345-377.
- [4] J. Yen, R. Langari, “Fuzzy Logic – Intelligence, Control, and Information”, Prentice Hall, 1999.
- [5] D. Thomas, B. Armstrong-Helouvy, “Fuzzy Logic Control – A Taxonomy of Demonstrated Benefits”, Proceedings of the IEEE, Vol. 83, Issue 3, March 1995, pp 407-421.
- [6] C. C. Lee, “Fuzzy logic in control systems: Fuzzy logic controller–Part I”, IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, Issue 2, Mar./Apr. 1990, pp 404-418.

- [7] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller–Part II", IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, Issue 2, Mar./Apr. 1990, pp 404-418.
- [8] Zhimin Ding, "Implementing fuzzy logic control with the XA", Philips Semiconductors, Application note AN710, Dec. 1996.
- [9] Shane Spiteri, "Embedded Fuzzy Control for Reefer Refrigeration Systems", Proceedings of 2001 IEEE International Fuzzy Systems Conference, 2001, pp 1088 – 1091
- [10] Y. K. Chen, C. H. Yang, Y. C. Wu, "Robust fuzzy controlled photovoltaic power inverter with Taguchi method", IEEE Transactions on Aerospace and Electronic Systems, Vol. 38, Issue 3, July 2002, pp 940- 954.
- [11] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, Prentice Hall, 1989.
- [12] J., Valente de Oliveira, J.M. Lemos, "A comparison of some adaptive-predictive fuzzy-control strategies", IEEE Transactions on, Systems, Man and Cybernetics, Part C, Vol. 30, Issue 1, Feb. 2000, pp 138-145.
- [13] L. Fortuna, M. Lo Presti, C. Vinci, A. Cucuccio, "Recent Trends in Fuzzy Control of Electrical Drives: an Industry Point of View", Proceedings of the 2003 International Symposium on Circuits and Systems, Vol. 3, May 2003, pp 459- 461.
- [14] D. Mihai, "Optimizing the Defuzzifier Timing for the Fuzzy Control of a Servodrive", Proceedings of 34th International Symposium on Multiple-Valued Logic, May 2004, pp 142- 147.
- [15] B. Giacalone, M. Lo Presti, F. Di Marco, "Hardware implementation versus software emulation of fuzzy algorithms in real applications", Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 1998, Vol. 1, pp 7-12.
- [16] R. Dapos Amore, O. Saotome, K.H. Kienitz, "A two-input, one-output bit-scalable architecture for fuzzyprocessors", IEEE Design & Test of Computers, Vol. 18, Issue 4, Jul/Aug 2001, pp 56-64.
- [17] D. Falchieri, A. Gabrielli, E. Gandolfi, M. Masetti, "Design of very high speed CMOS fuzzy processors for applications in high energy physics experiments", Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, Granada, Spain, April 1999, pp 284-291.
- [18] I. Baturone, A. Barriga, S. Sanchez-Solano, J.L. Huertas, "Mixed-signal design of a fully parallel fuzzy processor", Electronics Letters, Mar 1998, Vol. 34, Issue 5, pp 437-438.

## Knowledge-based Tuning I: Design and Tuning of Fuzzy Control Surfaces with Bezier Function

Hanqi Zhuang and S. Wongsoontorn

### 4.1 Introduction

Fuzzy logic controllers (FLCs) have rapidly gained popularity in many engineering disciplines after the concept of fuzzy logic was introduced by Zadeh [1] and after Mamdani and his coworkers [2] presented a fuzzy control scheme. In contrast to the conventional control theory, the design of FLCs is not solely based on the mathematical model of the process. Instead it incorporates human knowledge that can be mapped into membership functions and a set of control rules. Since the inception of FLCs, investigating effective tuning methods for FLCs has been an active research area. FLCs can be tuned by scaling control gains, modifying membership functions, adapting controller rules and adjusting entries of look-up tables. Mann *et al.* [3] proposed a two-stage tuning method for fuzzy PID controllers. First, a nonlinear tuning by changing membership functions is performed to deal with unknown nonlinearity such as dead time and hysteresis. Second, a linear tuning is done by adjusting gains of the fuzzy system. Their simulations showed improvement of the system in term of the step response over conventional systems. Li and Lau [4] presented a coarse-and-fine tuning method, in which one of two look-up tables is selected to control the plant based on the magnitude of errors. A tuning method using the concept of cell mapping was introduced by Smith and Comer [5]. A set of fuzzy control rules can be calibrated automatically by mapping the nonlinear input-output relationship. The input-output data is used to fine tune the output function parameters of a FLC through the use of Mean Square Error (MSE) estimation algorithm. The method of comparative gain design was proposed by Li [6], in which the gain of a well-tuned linear counterpart is used as the initial fuzzy gain of the FLC being tuned. After a sequential fine tuning stage by trial and error, the performance of the FLC could be further improved. Homaifar and McCormick [7] examined the applicability of Genetic Algorithms (GAs) in the simultaneous design of membership functions and rule sets for FLCs. Jantzen [8] studied four types of control surfaces, which indicated that the shapes of control surfaces had significant impact on the step responses of control systems. Bai *et al.* [9] proposed a tuning method to suppress noise in the laser tracking system in which scale factors were used to obtain multiple membership functions from only one membership function.

To use FLCs more efficiently, the tunable parameters (scaling factor, membership functions) need to be determined. There is no standard way to select a) the shapes of membership functions, b) the number of linguistic values, c) a universal rule base, and d) the most appropriate inference mechanism and defuzzification method. In this chapter we devise an alternative method to tune FLCs without much knowledge of tunable fuzzy parameters. A  $\mu$ -Law tuning method was first introduced by Zhuang and Wang [10], by which the fuzzy control surface was directly modified to achieve better control performance. This was made possible by modifying the control surface in term of the  $\mu$ -Law parameters. An advantage of the  $\mu$ -Law tuning method was its compactness. Very few parameters were used to model the fuzzy control surface. On the other hand, since there is only one parameter in the  $\mu$ -Law function, the flexibility of the surface is limited.

In the scheme proposed in this chapter, the Bezier function is used as the input-output mapping for an FLC. Thus, the control surface can be changed directly by adjusting the set of Bezier parameters. As more parameters are involved in the Bezier functions, the controller designed with the Bezier function can achieve better performance in comparison with the  $\mu$ -Law base FLC. A Genetic Algorithm (GA) is then applied to search for optimal parameters based on system performance indexes such as rise time, overshooting, setting time, steady state error, and integral time absolute error (ITAE), greatly reducing the need for human input in the design of a closed loop control system.

The remainder of the chapter is organized as follows. Basic control structures of FLCs, control surfaces and Bezier function are overviewed in Section 4.2. A tuning method based on GAs is outlined in Section 4.3. In Section 4.4, simulation results are given. The chapter ends with conclusions in Section 4.5.

## 4.2 Control Surfaces

The input-output relationship of a simple PD controller can be expressed mathematically as

$$U(s) = K_e E(s) + K_{\dot{e}} s E(s) \quad (4.1)$$

In the discrete time domain, the input-output relationship is described by

$$u(k) = K_e e(k) + K_{\dot{e}} \dot{e}(k) \quad (4.2)$$

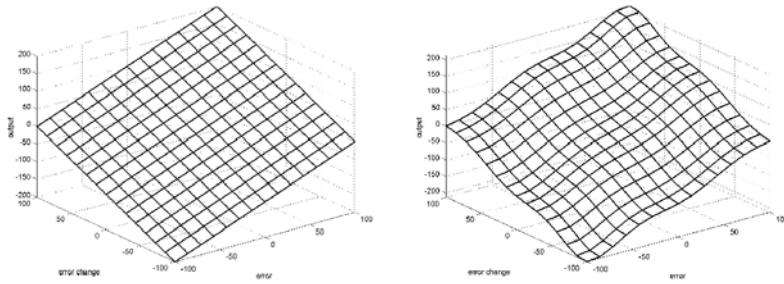
where the error  $e(k)$  and the error change  $\dot{e}(k)$  are the inputs of the controller and  $u(k)$  is the output of the controller, and  $K_e$  and  $K_{\dot{e}}$  are gains of the system. The error state variables are defined as  $e(k) = y(k) - r(k)$ ,  $\dot{e}(k) = e(k) - e(k-1)$ . The control output, process output and the set-point are denoted as  $u$ ,  $y$  and  $r$ , respectively.

In an FLC, the summation in linear conventional controller is replaced by a fuzzy controller acting like summation. With two inputs and one output the input-

output mapping is a surface. The functional relationship of a process controller can be written as

$$u(k) = f(e(k), \dot{e}(k)) \quad (4.3)$$

where  $f$  is a nonlinear function of  $e$  and  $\dot{e}$ , obtained by using membership function, controller rules, the inference engine and defuzzifications. The control surface of Equation (4.3) can be realized with a two-dimensional rule-base graphically represented in Figure 4.1. The linear surface in Figure 4.1a is a representation of a conventional PD controller. On the other hand, the surface depicted in Figure 4.1b is a typical FLC control surface. It is bumpy because it is a nonlinear mapping. The fact that the surface in Figure 4.1a can be viewed as a special case of that in Figure 4.1b suggests that the performance of an FLC should be no worse than its conventional counterpart if both are optimally tuned.



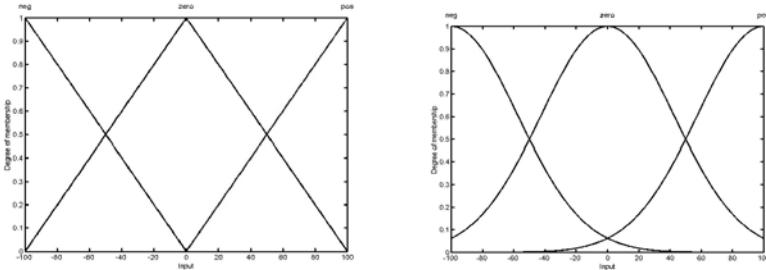
**Figure 4.1.** (a) Linear control surface (b) Bumpy control surface

How to choose a proper function to represent control surface is one of the tasks of this research. In Table 4.1 [8], the example fuzzy rules for computation of output  $u$  are used to produce control surface in Figure 4.1a and 4.1b. The **AND** connective is multiplication. The input membership functions as shown in Figure 4.2a and 4.2b are used for linear surface and bumpy surface respectively.

From this example of fuzzy rule base, one can see that the same rule base with different shapes of membership function can produce different control surfaces

**Table 4.1.** Fuzzy rule base

$\Delta e/e$	Negative	Zero	Positive
<b>Negative</b>	-200	-100	0
<b>Zero</b>	-100	0	100
<b>Positive</b>	0	100	200



**Figure 4.2.** (a) Triangle membership function (b) Gaussian membership function

Design and tuning of FLCs traditionally take two approaches. In the first approach, an FLC designed with the control rule in mind is tuned by adjusting rule-base parameters of the fuzzy system. In the second approach, tuning is performed by altering gains to obtain a better overall performance. While the first method may produce controllers with a superb performance, errors can occur during the membership mapping processes, as the control surface in both cases are realized using Centroid defuzzification in Equation (4.4)

$$y = \frac{\int_{a \in R} a U(a) da}{\int_{a \in R} U(a) da} \quad (4.4)$$

where  $U$  is the membership function with its set given by  $R = \{a \mid U(a) \neq 0\}$ .

An alternative way of design and tuning the control surface is considered here by exploring functions used to represent direct input-output mapping. The corresponding control surface can be changed by adjusting the parameters of the functions. The candidate functions must have the following properties: (1) flexible enough to allow sufficient variations of control surface, (2) simple and few parameters to facilitate a quick tuning.

Bézier curves are very useful for geometric modeling [11]. The curve shape can be controlled in a predictable way through changing a number of parameters. Bézier curves can be represented as follows:

$$p(e) = \sum_{i=0}^n p_i B_i, \quad e \in [0,1] \quad (4.5)$$

where

$$B_i, n = C(n, i) e^i (1 - e)^{n-i} \quad (4.6)$$

and  $C(n, i)$  is a binomial coefficient

$$C(n, i) = \frac{n!}{i!(n-i)!} \quad (4.7)$$

From Equation (4.5), one can obtain a polynomial form with  $n$ -points. In this chapter, three points,  $n=3$ , are investigated. That is,

$$p(e) = (1-e)^2 p_0 + 3e(1-e)^2 p_1 + 3e^2(1-e)p_2 + e^3 p_3 \quad (4.8)$$

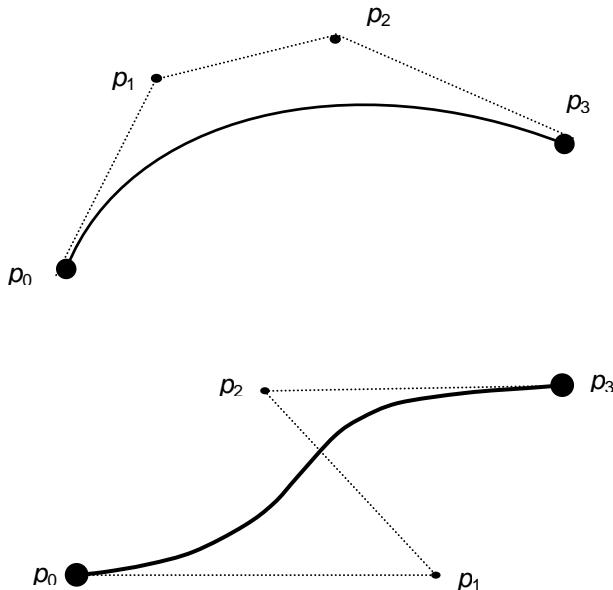
where the  $p_i$  represent the  $n+1$  vertices of a characteristic polygon. These vertices are also called control points.

Figure 4.3 shows two examples of Bezier curves. One can see that  $p_1$  and  $p_2$  are control points used for blending the curve while  $p_0$  and  $p_3$  are the starting and ending points of the curve. In this application, the curve is required to start from 0 and end at 1, and therefore Equation (4.8) becomes

$$p(e) = 3e(1-e)^2 p_1 + 3e^2(1-e)p_2 + e^3, \quad e \in [0,1] \quad (4.9)$$

For the FLCs, the universe of discourse of each input is defined to be within the range  $[-1, 1]$  as shown in Figure 4.4. A modification can be made for the negative side of the input, which can be represented as follows:

$$p(e) = 3e(1+e)^2 p_1 - 3e^2(1+e)p_2 + e^3, \quad e \in [-1,0] \quad (4.10)$$

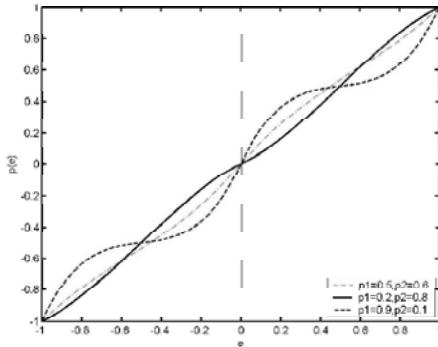


**Figure 4.3.** Bezier curve with three vertices

Once inputs are normalized to  $[-1, 1]$ , a gain  $K$  is introduced to handle the scaling issue:

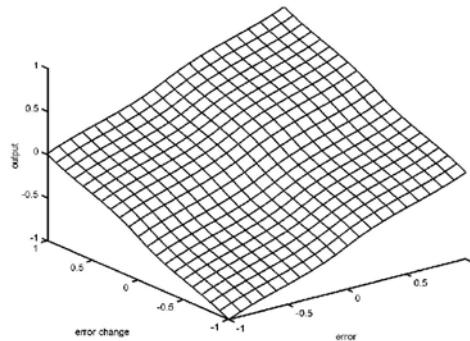
$$p(e) = K(3e(1-e)^2 p_1 + 3e^2(1-e)p_2 + e^3), \quad e \in [0,1] \quad (4.11)$$

$$p(e) = K(3e(1+e)^2 p_1 - 3e^2(1+e)p_2 + e^3), \quad e \in [-1,0] \quad (4.12)$$



**Figure 4.4.** Normalized Bezier curves ( $-1 < e < 1$ ) with various parameters  $p_1$  and  $p_2$

The control surface can be obtained from the summation of the two inputs, both governed by Equations (4.11) and (4.12). As an example, a surface with various parameters  $p_1=0.5$  and  $p_2=0.6$  on both two inputs is shown in Figure 4.5.



**Figure 4.5.** A surface implemented from summation of two Bezier functions with parameters  $p_1=0.5$  and  $p_2=0.6$

### 4.3 Tuning with a Genetic Algorithm

A simple genetic algorithm SGA [11] is composed of three main components: 1) initialization; 2) evaluation; and 3) genetic operations.

1) *Initialization*: In the SGA, a possible solution for the optimization problem is called an individual. A set of individuals represents a group of candidate solutions called a population. A GA begins with an initial population and thereafter generates better ones to find the best solution. Each individual is usually represented by a binary vector although other coding methods have been given in the literature [12].

2) *Evaluation*: Each individual is evaluated to obtain its fitness function, which is problem specific. The fitness function evaluates the relative fitness of a certain individual among a population.

3) *Genetic operations*: Three genetic operators, called selection, crossover, and mutation, are used. These operators are applied to selected individuals:

- **Selection**: This operator selects an individual from the current population to produce offsprings in the next one. The probability of an individual to be selected is proportional its fitness value.
- **Crossover**: This operator exchanges portions between selected individuals. It recombines the genetic material of two parent individuals to produce two offspring. The individuals are broken randomly at the same places and combined together to create two new offspring individuals.
- **Mutation**: This operator causes intermittent and random alteration of binary bits (changing zero to one and *vice versa*) of an individual string

Next, let's discuss the tuning parameters. Initially, these parameters must be selected and then coded, which includes setting the individual parameter length and generating genetic string (chromosomes) using coding strategies. As mentioned earlier, the gains and surface shape of FLC vary as the parameters of Bezier curves change in the tuning process. The tuning parameters can be described as follows:

$$(K_e, K_{\dot{e}}, K, p_{e1}, p_{e2}, p_{\dot{e}1}, p_{\dot{e}2}) \quad (4.13)$$

where  $K_e$ ,  $p_{e1}$  and  $p_{e2}$  are Bezier curve parameters associated with the fuzzy error term,  $K_{\dot{e}}$ ,  $p_{\dot{e}1}$  and  $p_{\dot{e}2}$  are the parameters associated with the fuzzy error change-term (refer to Equations (4.11) and (4.12)), and  $K$  is the output gain. The optimization problem is to find  $K_e, K_{\dot{e}}, p_{e1}, p_{e2}, p_{\dot{e}1}, p_{\dot{e}2}$  and  $K$  such that an objective function  $J$  is minimized.

The next step is to initialize SGA parameters including the size of the mating pool, crossover probability ( $P_c$ ) and mutation probability ( $P_m$ ). Once SGA parameters are initialized, a set of original chromosomes is generated randomly.

The fuzzy surface can then be constructed by using the chromosomes in the mating pool. The fitness of each parameter can be evaluated according to the fitness function. The generalized optimization problem for the controller is described as

$$\text{maximize } F = \frac{1}{J} \quad (4.14)$$

subject to

$$0 < (K_e, K_{\dot{e}}) \leq 1$$

$$\begin{aligned} 0 < (K) &\leq \max(|u_{\min}|, |u_{\max}|) \\ -2 &\leq (p_{e1}, p_{e2}, p_{\dot{e}1}, p_{\dot{e}2}) \leq 2 \end{aligned}$$

where  $F$  is the fitness function.  $K_e$  and  $K_{\dot{e}}$  gains are normalized in the range of 0 to 1 and  $K$  is a denormalized output gain to adjust the output control in the allowable range of the input plant. The values of  $p_{e1}, p_{e2}, p_{\dot{e}1}, p_{\dot{e}2}$  are limited to keep the Bezier curves staying in the range of -1 to 1. The fitness function is the summation of weighted measuring performance indexes:

$$\begin{aligned} J(K_e, K_{\dot{e}}, K, p_{e1}, p_{e2}, p_{\dot{e}1}, p_{\dot{e}2}) &= \\ \sum_{i=1}^n x_i J_i(K_e, K_{\dot{e}}, K, p_{e1}, p_{e2}, p_{\dot{e}1}, p_{\dot{e}2}) & \end{aligned} \quad (4.15)$$

where  $x_i$  is a weight associated with the  $i$ th individual measuring performance index. The weight can be adjusted according to which performance index we want to emphasize over others. The study in [12] shows that a single index can lead to poor tuning performance. Here, four measuring performance indexes are used in this research:

$$J = \left( x_1 \frac{T_r}{T} + x_2 \frac{T_s}{T} + x_3 OS + x_4 IAE \right) \quad (4.16)$$

where  $T_r$ ,  $T_s$ ,  $T$ ,  $OS$  and  $IAE$  are defined as rise time, setting time, simulation time, overshoot, and integral absolute error respectively. From Equation (4.16), the smaller the value of  $J$ , the better the performance of the FLC, which means that a high value of  $F$  corresponds to a relatively good performance.

After evaluating the fitness function, new offspring will be generated following the three steps in the GA: selection, crossover, and mutation. The genetic operators used in this work are uniform crossover, bit mutation, and roulette wheel selection [12].

Finally, the strings in the mating pool are reproduced by the offspring. The above procedure is repeated until the output satisfied the desired criteria or certain terminal conditions are satisfied. After many experimental runs for tuning the GA the following parameter values were selected in our study: population size = 100; crossover probability = 0.8; mutation probability = 0.1; and number of bits of parameters = 10.

## 4.4 Simulation Results

The simulation results presented in this section are typical third-order, or fourth-order oscillatory processes, or a third-order plant with a long delay time. Using the proposed Bezier Curve Fuzzy PI controller (BCFPI) as shown in Figure 4.6, the performances of the proposed scheme are compared with those of conventional rule base fuzzy PI controllers. In order to make a comparison between the conventional and proposed fuzzy PI controllers, a combination of measuring performances such as rise time ( $T_r$ ), setting time ( $T_s$ ), overshoot ( $OS$ ), integral absolute error (IAE), and integral of time absolute error (ITAE) is used, as discussed in Section 4.3. The three process tests have been chosen because these were known to be difficult to control with a PID controller. All of the three case simulations are compared with the Linear Fuzzy PI (LFPI) [8] using gains that show in Table 4.2 and fuzzy rules in Table 4.1.

**Table 4.2.** Gains setting for linear fuzzy PI (LFPI) in all processes

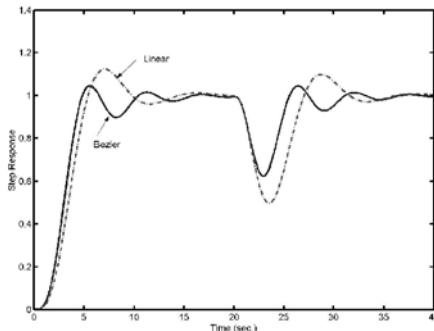
Gain	$G_1(s)$	$G_2(s)$	$G_3(s)$
$K_e$	100	15	100
$K_e$	150	120	400
$K$	0.006	0.00125	0.001

The plant models  $G_1(s)$ ,  $G_2(s)$  and  $G_3(s)$  will be given shortly. The simulation is conducted in MATLAB environment utilizing Simulink and the Fuzzy Logic Toolbox to implement the rule bases. For the numerical integral, the fourth-order Runge-Kutta method with interval of 0.1 s. is used for all processes. After the GA terminated the parameters of BCFPI were obtained, which are shown in Table 4.3.

The first plant is a third-order linear process with the transfer function  $G_1(s) = 1/(s+1)^3$ . Figure 4.6 shows the step responses of both BCFPI and LFPI. A load disturbance is applied at  $t = 20$  s. It is evident from the figure that BCFPI exhibits a very good performance against load disturbance compared with LFPI. Table 4.4 reveals a numerically superior performance of BCFPI over LFPI.

**Table 4.3.** The parameters of BCFPI

Parameters	$G_1(s)$	$G_2(s)$	$G_3(s)$
$K_e$	0.370	0.332	0.043
$K_e$	0.621	0.5197	0.6278
$K$	1	0.18	1.5
$p_{e1}$	0.456	0.0432	0.4691
$p_{e2}$	1.57	0.8575	0.6142
$p_{\dot{e}1}$	1.07	1.315	0.0196
$p_{\dot{e}2}$	1.09	0.1846	1.573

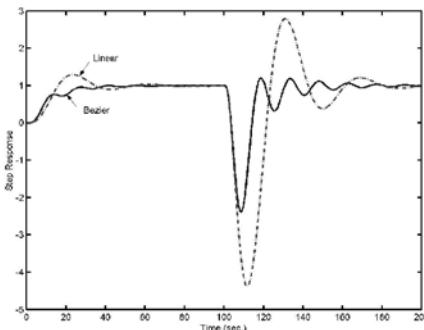


**Figure 4.6.** Step responses of BCFPI and LFPI for  $G_1(s)$

**Table 4.4.** Performance comparison for  $G_1(s)$

FLC	%OS	$T_r$	$T_s$	IAE	ITAE
<b>BCFPI</b>	4.488	2.68	14.6	4.73	42.17
<b>LFPI</b>	12.273	3.06	13.5	6.12	66.85

The second case is a system with its transfer function  $G_2(s) = 1/s(s+1)^3$ . It possesses oscillatory modes because one of its pole is at the origin ( $s = 0$ ). The system is very difficult to dampen because it has more than one oscillatory mode. As an illustration, Figure 4.7 and Table 4.5 show in comparison the performances of BCFPI and LFPI. Again, BCFPI outperformed LFPI for both overshoots and load disturbance.

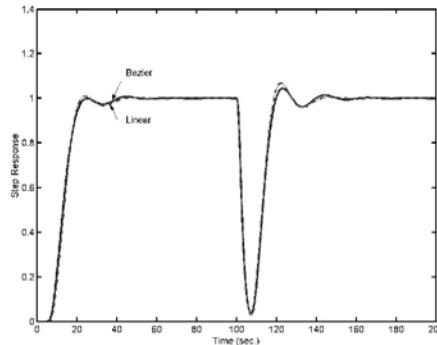


**Figure 4.7.** Step responses of BCFPI and LFPI for  $G_2(s)$

**Table 4.5.** Performance comparison for  $G_2(s)$

FLC	%OS	$T_r$	$T_s$	IAE	ITAE
<b>BCFPI</b>	0.78	19.412	50.91	50.2	4450
<b>LFPI</b>	67.68	9	67.68	116.38	12450

The third is a system with a very large dead time. The transfer function described as  $G_3(s) = e^{-5s} / (s+1)^3$  has a significant delay time of 5 seconds. Systems with a long delay time are difficult to control. Table 4.6 shows the quantitative performance of BCFPI and LFPI. From the results, it is evident that BCFPI is better than LFPI in terms of IAE and ITAE.



**Figure 4.8.** Step responses of BCFPI and LFPI for  $G_3(s)$

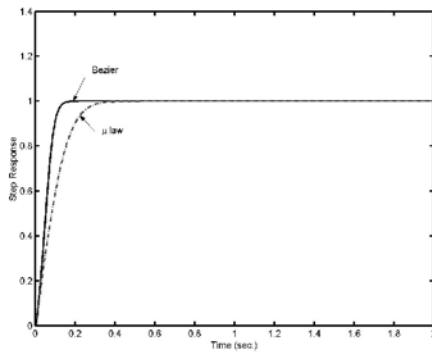
**Table 4.6.** Measuring performance comparison for  $G_3(s)$

FLC	%OS	$T_r$	$T_s$	IAE	ITAE
<b>BCFPI</b>	0.708	10.51	36.2	24.6	1314
<b>LFPI</b>	0.856	9.71	38.11	25.6	1325

Finally a laser tracking system [9] was used to test the performance of Bezier curve FLC by comparing the results with the  $\mu$ -Law FLC [10]. The system was identified as  $G(s) = 10.68 \frac{s + 610.5}{s(s + 430.6)} e^{-0.005s}$ . The time delay of 5 ms was attributed to the communication time between the computer controller and the motor gimbals. The test results are shown in Figure 4.9 and Table 4.7.

**Table 4.7.** Performance comparison for a laser tracking system

FLC	%OS	$T_r$	$T_s$	IAE
<b>Bezier</b>	0.0029	0.0815	0.131	0.0566
<b><math>\mu</math>-Law</b>	0.000732	0.1814	0.288	0.0985



**Figure 4.9.** Step responses of Bezier curve and  $\mu$ -Law FLC

From Table 4.7, it can be seen that, in most aspects, the Bezier curve FLC outperforms the  $\mu$ -Law FLC.

## 4.5 Conclusions

In this chapter, a new design and tuning scheme is proposed for FLCs. The Bezier curve function is used to adjust surface shape by means of varying a set of parameters without the knowledge of traditional FLC parameters (scaling factors, membership functions, and control rules). The proposed tuning method can be applied to various forms of FLCs as the surface shape of these controllers can be readily presented by Bezier functions. In the simulation study, the GA was applied to adjust automatically parameters of Bezier functions and therefore control surfaces of FLCs. Simulation results showed that the proposed design and tuning scheme could significantly improve the overall performance of FLCs.

The fuzzy surfaces obtained by the method proposed in the chapter can be mapped to membership functions, with which one can implement traditional fuzzy logic controllers.

## References

- [1] L.A. Zadeh, "Fuzzy sets", *Information and Control*, Vol. 8, pp 338-353, 1965.
- [2] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc. IEE Contr. Sci.*, Vol. 121, pp 1585-1588, 1974.
- [3] G. M. I. Mann, B. -G. Hu, and R. G. Gosine, "Two level tuning of fuzzy PID controllers," *IEEE Trans. Syst., Man, Cybern. B*, Vol. 31, pp 263-269, 2001.
- [4] H.-X. Li and C. C. Lau, "Development of fuzzy algorithms for servo systems," *IEEE Contr. Syst. Mag.*, pp 65-71, Aug. 1989.

- [5] S. M. Smith and D. J. Comer, "Automated calibration of a fuzzy logic controller using a cell state space algorithm," *IEEE Contr. Syst. Mag.*, pp 12-28, Aug. 1991.
- [6] H.-X. Li, "A comparative design and tuning for conventional control," *IEEE Trans. Syst., Man, Cybern. B*, Vol. 27, pp 884-889, 1997.
- [7] A. Homaifar and E. McCormick, "Simultaneous design of membership functions and rule sets controllers using genetic algorithms," *IEEE Trans. Syst. Man, Cybern.*, Vol. 3, pp 129-138, 1995.
- [8] J. Jantzen, "A robustness study of fuzzy control rules," in EUFIT (ed), *proceedings Fifth European Congress on Fuzzy and Intelligent technologies*, pp 1222-1227, 1997.
- [9] Y. Bai, H. Zhuang, Z. S. Roth, "Fuzzy Logic Control to Suppress Noises and Coupling Effects in a Laser Tracking System", *IEEE Trans on Control Systems Technology*, Vol. 13, No.1, January 2005, pp 113-121.
- [10] H. Zhuang and Y. Wang, " $\mu$ -Law tuning of a fuzzy lookup table," *Intelligent Automation and Soft Computing*, Vol. 8, No. 1, pp 1-14, 2001.
- [11] M. E. Mortenson, *Geometric Modeling*. Wiley, 1985.
- [12] D. E. Goldberg, *Genetic Algorithms in Search Optimization & Machine Learning*. Addison Wesley Longman, Inc. 1989.

## Knowledge-based Tuning II: $\mu$ -Law Tuning of a Fuzzy Lookup Table

Hanqi Zhuang and Yingli Wang

### 5.1 Introduction

Motivated by the effort due to Zadeh on fuzzy set theory [1-3], Mamdani *et al.* initiated fuzzy control schemes [4-11]. Since its inception, the concept of Fuzzy Logic Controllers (FLCs) has been successfully applied to control systems which are difficult to model due to nonlinearity and other complexities. Active research issues on FLCs include controller design, stability analysis, performance analysis and associated parameter tuning. The present study is devoted to tuning FLCs.

There are a number of methods proposed for tuning FLCs in the literature [12-17]. They can be classified into tuning of the membership functions, controller rules and lookup tables. For example, Li and Lau [12] proposed a method by which two lookup tables (coarse and fine) can be designed and tuned. A tuning method based on the concept of cell mapping was introduced by Smith and Comer. Rather than performing tuning of a lookup table, a method of adjusting scaling factors was proposed by Li and Gatland, in which the gains of a well-tuned linear counterpart are used as the initial fuzzy gains of the FLC being tuned.

In this chapter, a new tuning method that changes the structure of a fuzzy lookup table through varying a set of parameters is introduced. In this approach, the entries of the fuzzy lookup table are obtained by sampling a function defined in terms of a set of the so-called  $\mu$ -Law parameters. By adjusting these parameters, the control structure of the FLC is changed. In turn, the performance of the FLC is optimized by adjusting the  $\mu$ -Law parameters using an optimization algorithm. The proposed method has been tested with appropriate experimental studies.

### 5.2 Gain and Surface Tuning of a Fuzzy Lookup Table

As mentioned earlier, this research is focused on modifying scaling gains and the surface of a lookup table in an FLC by means of varying a set of the  $\mu$ -Law parameters. Intuitively, it can be expected that the resulting lookup table will have a better control performance if both its scaling gains and surface shape are tuned. The following is a simple example that illustrates the underlying concept of gain and surface tuning.

### 5.2.1 PD Controller and Fuzzy PD Controller

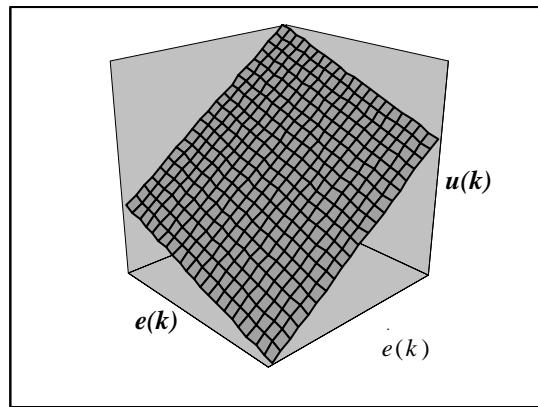
The control law of a linear PD controller is given by

$$U(s) = K_P E(s) + K_D sE(s) \quad (5.1)$$

with a corresponding input-output relationship in the discrete-time domain specified by

$$u(k) = K_P \cdot e(k) + K_D \cdot e(k) \quad (5.2)$$

Here the error  $e(k)$  and the error change  $e(k)$  are the inputs of the PD controller, and  $u(k)$  denotes the output of the controller. The constants of the system are  $K_P$  and  $K_D$ . The relevant input-output relationship of the PD controller is depicted by an inclined plane shown where the horizontal axes represents  $e(k)$  and  $e(k)$ , and the vertical axis represents  $u(k)$ . The shape of inclined plane is determined in terms of  $K_P$  and  $K_D$ .

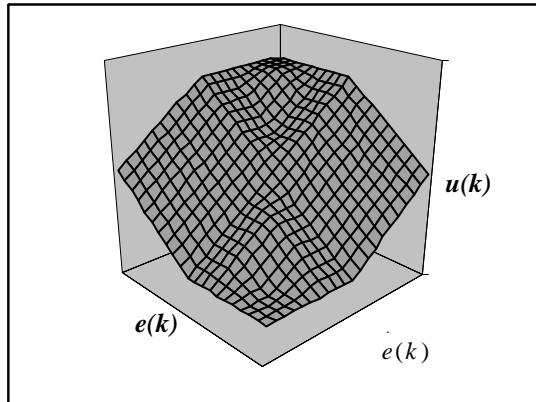


**Figure 5.1.** Input-output relationship of the classical PD controller

The performance of the PD controller,  $u(k)$ , may be improved by using a more complex function as follows:

$$u(k) = f \left[ e(k), e(k) \right] \quad (5.3)$$

where  $f$  is an arbitrary function which can be approximated by a piece-wise linear function of two variables. Using membership functions and appropriate controller rules, the fuzzy PD controller of Equation (5.3) can be realized with a two-dimensional lookup table graphically represented in Figure 5.2. The surface in Figure 5.2 is a folded plane as a result of piecewise approximation.



**Figure 5.2.** Input-output relationship of fuzzy PD controller

The inclined plane modeled in Figure 5.1 is a special case of a folded plane. In other words, the linear PD controller (an inclined plane) is a special case of a fuzzy PD controller (a folded plane). Therefore, a well-tuned fuzzy PD should normally possess better performance than a linear PD. To obtain a high performance fuzzy PD controller, both the input ranges and surface shape of the lookup table must be adjusted or tuned.

### 5.2.2 $\mu$ -Law Tuning of Fuzzy Lookup Table

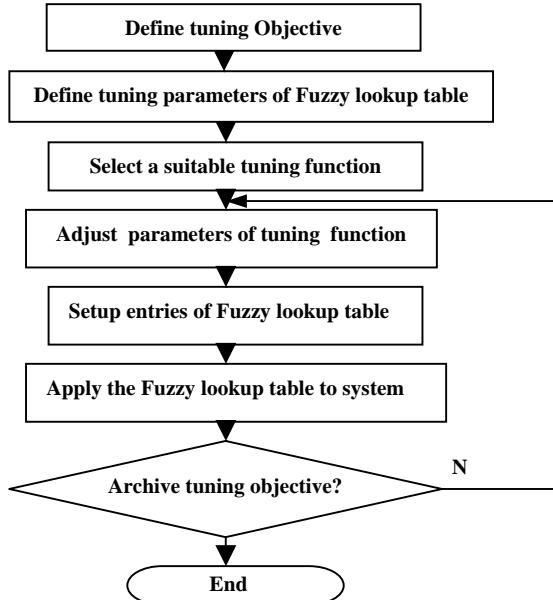
A simple method to tune the fuzzy lookup table is to adjust only the scaling gains and ignore the shape adjustment. On the other hand, one may vary also every entry of the fuzzy lookup table. While the latter method may produce a controller with very good performance, the tuning process is computationally too expensive in practice due to the fact that a large number of parameters need to be varied.

As such an alternative way of adjusting the surface of the lookup table can be considered. Here a function that represents the shape of the lookup table is used. Tuning the lookup table is then equivalent to adjusting the parameters of this function. Criteria to select a suitable function for such an application are: (1) it should be flexible enough to allow a rich variation of control surfaces; and, (2) it should be as simple as possible to facilitate a quick tuning of its parameters.

This process is illustrated in Figure 5.3.

How to choose a proper control surface function for tuning of FLCs is an interesting research issue that has both theoretical and practical importance. In this research, a particular tuning function with the intention to reduce the complexity of

the tuning procedure is indicated. An example of a fuzzy lookup table is presented in Table 5.1.



**Figure 5.3.** Flow chart of fuzzy lookup table tuning

**Table 5.1.** An example of a fuzzy PD lookuptable

$e \backslash e$	-3	-2	-1	0	1	2	3
-3	-10.2	-9.7	-8.9	-4.8	-0.8	0.1	0.6
-2	-7.0	-6.5	-5.7	-1.6	2.4	3.3	3.8
-1	-5.8	-5.3	-4.5	-0.4	3.6	4.5	5.0
0	-5.4	-4.9	-4.1	0	4.0	4.9	5.4
1	-1.6	-1.1	-0.3	3.8	7.8	8.7	9.2
2	-1.0	-0.5	0.4	4.4	8.4	9.3	9.8
3	-0.6	-0.1	0.7	4.8	8.8	9.7	10.2

A crucial step in the tuning process is to design a tuning function. We achieve this by using a two-stage procedure. In the first stage, a  $\mu$ -Law function is proposed which is defined in the normalized range of  $[-1, 1]$  for a control input. In the second stage, a scaling gain is introduced to handle practical cases in which inputs are not normalized to  $[-1, 1]$ .

For simplicity, in the following discussion, we first only consider a fuzzy P controller, in which case a control surface is reduced to a control curve. Later, we will extend the result to a fuzzy PD controller.

### A. A $\mu$ -Law Function

In the single input and single output case, the  $\mu$ -Law function is described by

$$y = \frac{\ln(1 + \mu x)}{\ln(1 + \mu)}, \quad 0 < x < 1. \quad (5.4)$$

where  $x$  is the input,  $y$  is the output and  $\mu$  is the parameter. Note that both  $x$  and  $y$  are functions of time in a control system.

The function is sketched for a set of selected values of  $\mu$  in Figure 5.4.

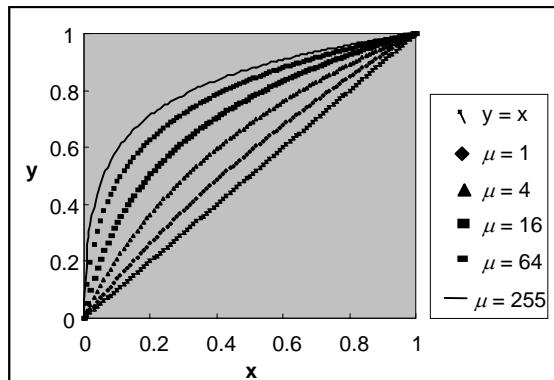


Figure 5.4.  $\mu$ -Law function

Since the function does not include the curves under the straight line  $y = x$ , a mirror  $\mu$ -Law function about  $y = x$  is defined as follows:

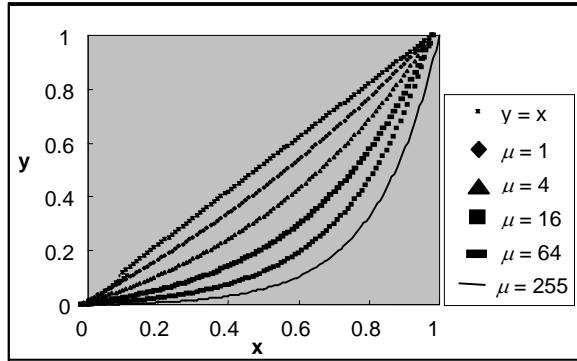
$$y = \frac{(1 + \mu)^x - 1}{\mu}, \quad 0 < x < 1. \quad (5.5)$$

The function for a selected set of values of  $\mu$  is shown in Figure 5.5.

Without loss of generality, the input to the controller is normalized to  $[-1, 1]$ . Relevant modified expression for  $y$  is as follows:

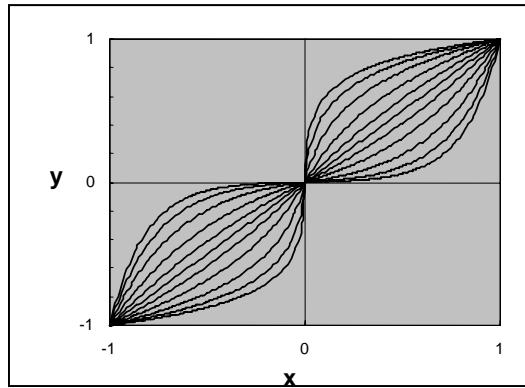
$$y = \text{Sign}(x) \frac{\ln(1 + \mu|x|)}{\ln(1 + \mu)}, \quad -1 < x < 1, \quad (5.6a)$$

$$y = \text{Sign}(x) \frac{(1 + \mu)^{|x|} - 1}{\mu}, \quad -1 < x < 1. \quad (5.6b)$$



**Figure 5.5.** Mirror of  $\mu$ -Law function about  $y=x$

The graphical representation of the  $\mu$ -Law tuning function for a range of values of  $\mu$  is plotted in Figure 5.6.



**Figure 5.6.** The  $\mu$ -Law tuning function

### B. Scaling

As has been mentioned, to accommodate inputs that are not normalized, a scaling factor has to be introduced. If one only considers scaling, the change of control gain will lead to the change of output range of the fuzzy lookup table with respect to the basic function  $u(k) = e(k)$ , as shown in Figure 5.7.

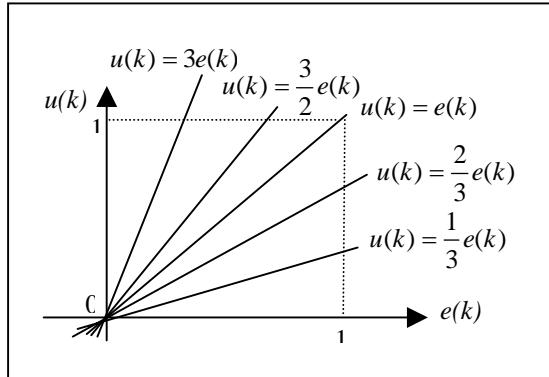


Figure 5.7. Input-output relationship when gain changes

### C. $\mu$ -Law Tuning of Fuzzy Lookup Table

By combining both shape and gain changes, more flexible curves can be obtained. The  $\mu$ -Law tuning algorithm of fuzzy lookup table introduced in this chapter may apply to nonlinear as well as linear systems. Typically, the ascending behavior of a nonlinear system is different from its descending behavior. Therefore, optimal control surfaces for a nonlinear system may not be symmetric. To achieve this, two different parameters  $\mu_1$  and  $\mu_2$  are used for two segments  $[-1,0)$  and  $[0,1]$ , respectively.

A modified  $\mu$ -Law function that allows both the gain and shape tuning can then be defined by Equation (5.7):

$$y = \begin{cases} K \frac{\ln(1 + \mu_1 x)}{\ln(1 + \mu_1)} & 0 < x < 1 \quad \text{and} \quad \mu_1 \geq 0 \\ -K \frac{(1 - \mu_1)^x - 1}{\mu_1} & 0 < x < 1 \quad \text{and} \quad \mu_1 \leq 0 \\ 0 & x = 0 \\ -K \frac{\ln(1 - \mu_2 x)}{\ln(1 + \mu_2)} & -1 < x < 0 \quad \text{and} \quad \mu_2 \geq 0 \\ -K \frac{1 - (1 - \mu_2)^{-x}}{\mu_2} & -1 < x < 0 \quad \text{and} \quad \mu_2 \leq 0 \end{cases} \quad (5.7)$$

where  $K$  is a scaling factor.

Note that when  $\mu_1$  and  $\mu_2$  approach zero, the controller defined in (5.7) becomes a linear P controller. By carefully selecting the values of  $\mu_1$  and  $\mu_2$ , one expects that the fuzzy P controller will out-perform a linear P controller.

This modified  $\mu$ -Law function can be directly applied to the tuning of a fuzzy P controller. Let the tuning parameters be  $K$ ,  $\mu_1$  and  $\mu_2$ . The objective of tuning of a fuzzy PD controller is to achieve an optimal control surface. Therefore, two

modified  $\mu$ -Law functions with six parameters altogether are applied to regulate the control variable in both the error and error rate directions. A similar tuning method can also be applied to a fuzzy PID controller provided that the controller is decoupled into some sort of fuzzy PD and fuzzy PI components.

### 5.3 Simulation and Experimental Results

In this section, simulation and experimental results obtained by applying the  $\mu$ -Law tuning method are presented. First, we present the conditions and results of simulation.

A fuzzy PD controller being tuned is represented via a lookup table. As indicated earlier, the gain and surface shape of the FLC varies as the parameters of the  $\mu$ -Law function change in the tuning process. The size of the fuzzy lookup table in the simulation is  $(7 \times 7)$ . Increasing the size of the lookup table does not increase the number of tuning parameters, which is one of the advantages of the  $\mu$ -Law tuning method proposed here.

The fuzzy parameter-tuning problem can be described by the following equation:

$$J = f(K_P, \mu_{1P}, \mu_{2P}, K_D, \mu_{1D}, \mu_{2D}) \quad (5.8)$$

where  $K_P$ ,  $\mu_{1P}$  and  $\mu_{2P}$  are  $\mu$ -Law parameters associated with the fuzzy *P*-term and  $K_D$ ,  $\mu_{1D}$  and  $\mu_{2D}$  are the parameters associated with the fuzzy *D*-term. The optimization problem is to find  $K_P$ ,  $\mu_{1P}$ ,  $\mu_{2P}$ ,  $K_D$ ,  $\mu_{1D}$  and  $\mu_{2D}$  such that the cost function  $f$  is minimized.

The FLC tuning process is considered here as a searching process. That is, a simplex search algorithm is used to optimize the cost function by selecting the parameters of the  $\mu$ -Law function, and in turn the performance of the fuzzy controller is optimized. Other optimization procedures such as simulated annealing or genetics algorithms can also be applied to the tuning problem. In a tuning process, the entries of the lookup table are initially filled with samples from a plane, which can be obtained from a linear PD controller. Then a predefined set of input signals are applied to the system, and outputs are evaluated based on a predefined objective function. A typical objective function is given in Equation (5.9). If the result meets certain terminal conditions, for instance, the objective function reaches a minimum point, the tuning process will stop. Otherwise, the parameters of the  $\mu$ -Law function are adjusted, and in turn, the corresponding entries of the lookup table are altered to seek a better control performance. The procedure is continued until a set of optimal parameters is obtained or a terminal condition (stop criterion) is satisfied. This tuning process is illustrated in Figure 5.8.

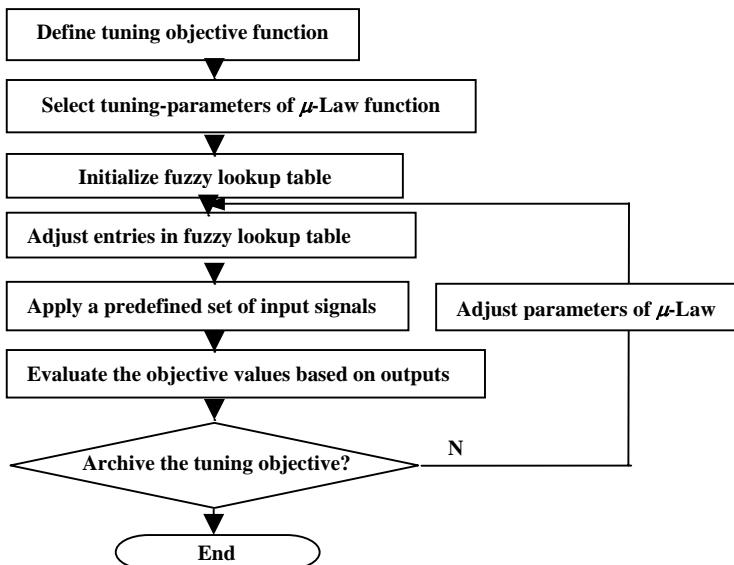
In the simulations performed, a third-order linear model given below was chosen to represent the plant to be controlled:

$$G(s) = \frac{1}{s^3 + 4s^2 + 4s + 4} \quad (5.9)$$

A unit feedback control system was assumed and the controllers were implemented with a 0.5 second time delay as shown in Figure 5.9. In the figure,  $R(s)$  and  $Y(s)$  are the input and output respectively,  $C(s)$  is the controller,  $G(s)$  refers to the plant, and  $D$  represents the time delay block. A unit step input signal was applied. The sampling rate was set to 0.1 second. Although other objective functions could be used, in the simulation, the following objective function was used:

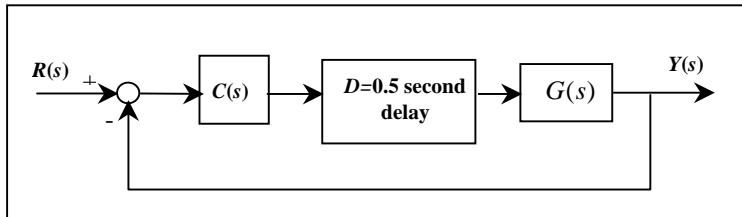
$$J = \alpha \frac{T_r}{T_{rs}} + \beta \frac{T_p}{T_{ps}} + \gamma \frac{T_s}{T_{ss}} + \delta \frac{OS}{OS_s} \quad (5.10)$$

where  $T_r$ ,  $T_p$ ,  $T_s$  and  $OS$  are defined as rise time, peak time, settling time and percent overshoot.  $T_{rs}$ ,  $T_{ps}$ ,  $T_{ss}$  and  $OS_s$  are reference constants of  $T_r$ ,  $T_p$ ,  $T_s$  and  $OS$ , respectively;  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\delta$  are the adjustment weights. The weighting coefficients associated with the terms in the cost function such as rise time, peak time, settling time, and the percent overshoot should be determined by the control objective. For example, for a laser tracking control system, in order to increase the tracking speed, rise time should be short. Also, as a result of the physical constraint of the optical sensors adopted in the laser tracking system, overshoot should be kept within a specified range so that the target will not be missed. In this simulation, the objective is to demonstrate that the proposed tuning method can be



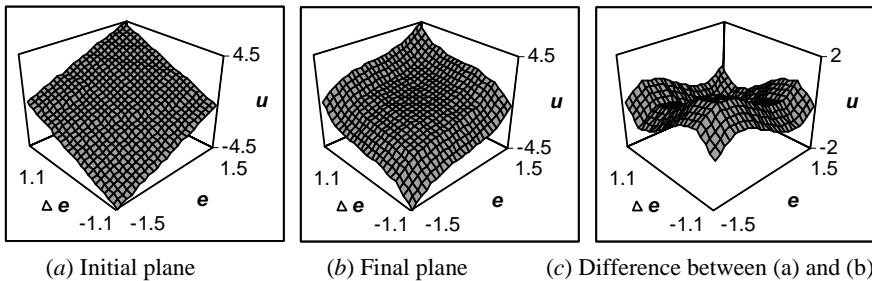
**Figure 5.8.** Flow chart of  $\mu$ -Law tuning of fuzzy lookup table

applied to control systems. Based on a manually tuned step response,  $T_{rs}$ ,  $T_{ps}$ ,  $T_{ss}$  and  $OS_s$  were assigned 0.7s, 1.0s, 2.5s, and 5%. Since rise time, peak time and overshoot were of secondary importance, more attention was paid to the overshoot and settling time. Hence, the weights of the rise time and peak time ( $\alpha = 0.25$  and  $\beta = 0.25$ ) were chosen to be much less than those of the settling time and overshoot ( $\gamma = 1$  and  $\delta = 1$ ).

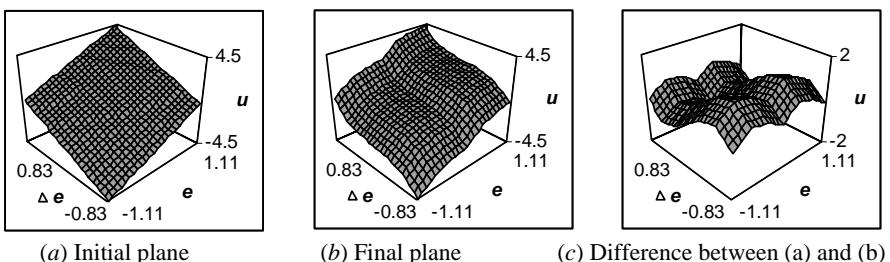


**Figure 5.9.** The system under study

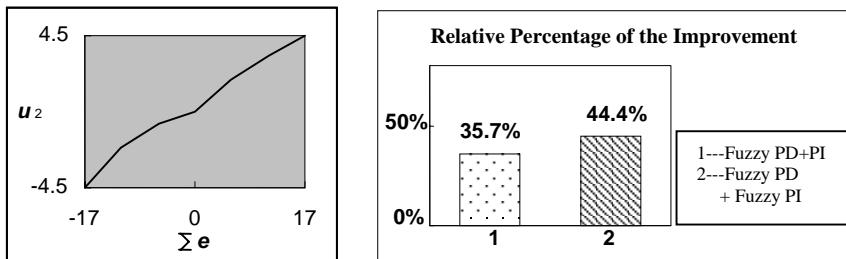
Figure 5.10 shows the tuned surface of a fuzzy PD controller lookup table, and Figure 5.11 shows that of a fuzzy PD plus fuzzy I controller. In the former case, a constant integral term was added to the fuzzy PD and the integral constant was tuned together with the fuzzy PD term. In Figures 5.10a and 5.11a, the control surface of a well-tuned linear PD controller is shown.



**Figure 5.10.** Final surface of simplex- $\mu$ -Law tuning of fuzzy PD controller



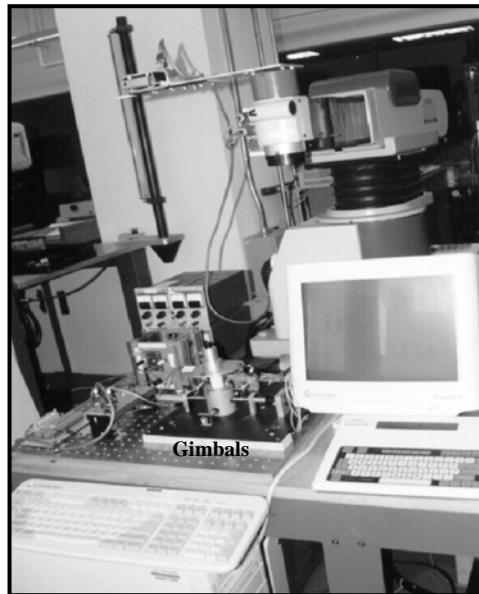
**Figure 5.11.** Final surface of simplex- $\mu$ -Law tuning of fuzzy PD+fuzzy I controller



(a) Final curve of integral (b)Relative percentage of the improvement

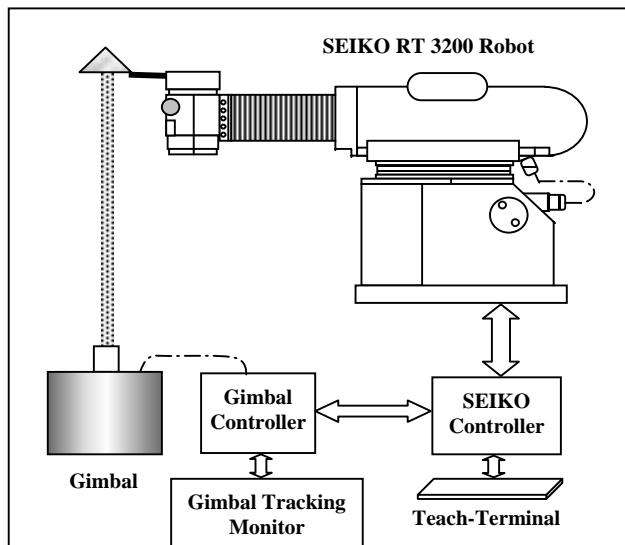
**Figure 5.12.** Final curve of integral and relative percentage of the improvement

Figures 5.10b and 5.11b show respectively the control surfaces of a well-tuned fuzzy PD and fuzzy PD+ fuzzy I controller. In the later case, the integral term was also fuzzified. Figure 5.12 shows the final curve of integral (a) and the percentage improvement of the control performance after and before fuzzy controllers are applied (b). The control performance is described by the value of the objective function given in (5.9) with parameters  $T_{rs} = 0.7s$ ,  $T_{ps} = 1.0s$ ,  $T_{ss} = 2.5s$ ,  $OS = 5$ ,  $\alpha = 0.25$ ,  $\beta = 0.25$ ,  $\gamma = 1$  and  $\delta = 1$ . More specifically, in Figure 5.12b, we compare a linear PD controller with two fuzzy controllers, all of which are optimized by the same downhill simplex algorithm.

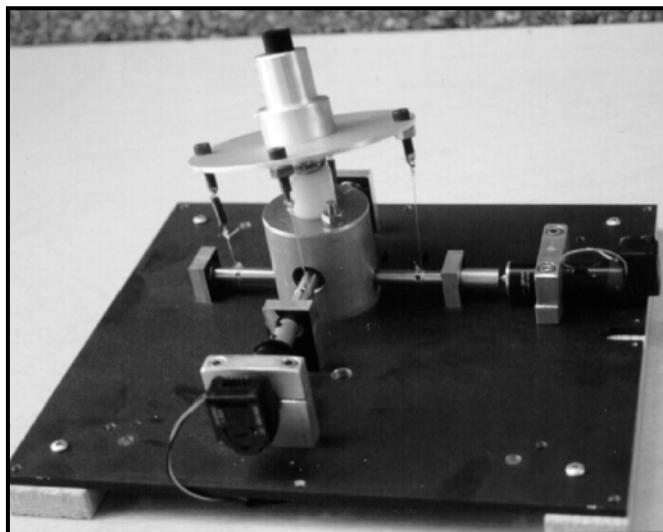


**Figure 5.13.** The experimental system setup in the FAU robotics center

To examine further the effectiveness of the proposed method, the  $\mu$ -Law tuning procedure was applied to the controller design for a laser tracking system (LTS) at the Robotics Center, Florida Atlantic University with the setup shown in Figure 5.13. The sketch of the LTS experimental environment is illustrated in Figure 5.14.



**Figure 5.14.** Sketch of the experimental system



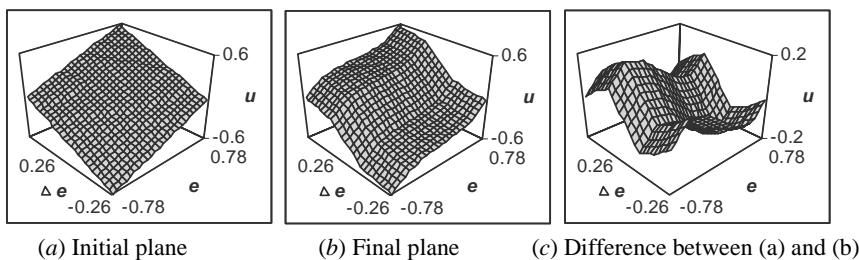
**Figure 5.15.** The laser tracking gimbal used for the experiment

The objective of the control system was to track a target mounted on the end-effector of a robot using a laser beam emitted by a laser pointer. In this system, a four-quadrant detector target was held by a SEIKO RT3200 robot, a four-axis, closed-loop DC servo manipulator.

A laser tracking gimbal on which the laser pointer was mounted (shown in Figure 5.15) was used to carry on the tracking task. The gimbal controller was set to collect data from the four-quadrant detector, sensing the deviation of the laser beam from its center. The resulting error signals were processed by a control algorithm to drive a servo system, which in turn adjusted the angles of the LTG so that the laser beam could reach the center of the four-quadrant detector.

In this laser tracking system, because of the physical size constraint of the four-quadrant detector, a critical issue was to maintain the tracking status without losing the target while tracking. Therefore, the main objective was to minimize the maximum tracking error at a high speed. Another objective was to minimize the steady-state error when the target remains still. An auto-tuning algorithm was adopted. Those parameters that lead to unstable controllers were immediately rejected. The selected set of parameters was the one that minimized tracking errors while maintained the stability of the plant.

An important issue for automatic tuning of control parameters was the auto-recovery of the tuning process resulting from poor choice of the control parameters. Once an optimal tuning algorithm was applied, during the searching process, one might not guarantee that all the resulting parameters were good enough to maintain target tracking. Some sets of the parameters might be bad and thus might lead to the loss of the target, which would interrupt the auto-tuning process. This problem, however, could be solved by the following technique. An initial position of the target was recorded through the encoder outputs. Whenever the target was lost, the tracking gimbal would go back to its original position and wait for the target. Further, the training trajectory was defined as a repeated line, and a pre-tuned set of control parameters was used to track the target once the target went back to the predefined point.



**Figure 5.16.** Final surface of simplex- $\mu$ -Law tuning of fuzzy PD controller (experiment)

The fuzzy lookup table was initialized by a set of manually tuned PD parameters. By adjusting their surface shape and gains through the downhill simplex algorithm, a fuzzy PD controller was obtained with its control surface shown in Figure 5.16b. When the target was 1 m away, the maximum tracking speed could reach 4.8 m/min, which was about a 40% improvement over the

performance compared with the PD controller whose control parameters ( $K_P$  and  $K_D$ ) are optimized by the same downhill simplex algorithm.

## 5.4 Summary

In this chapter, a new tuning method that can adjust flexibly both the scaling gain and surface shape of a fuzzy lookup table by means of varying a set of  $\mu$ -Law parameters is introduced. The proposed tuning technique can be applied to various forms of fuzzy PID controllers as the structures of these controllers can be readily represented by the  $\mu$ -Law parameters.

In the simulation and experimental studies, the simplex optimal algorithm was combined with the  $\mu$ -Law method to adjust automatically the parameters of the  $\mu$ -Law function and therefore the structures of the FLCs. Simulation and experimental results showed that the  $\mu$ -Law tuning method could improve significantly the performance of FLCs.

In comparison to the approach of tuning membership functions, the number of tuning parameters in the proposed method is small. In turn, this reduces the computational complexity of an optimal tuning algorithm. The trade-off in applying the  $\mu$ -Law function is the constraint of the surface shape, which means that even if the FLC is optimized in terms of the  $\mu$ -Law parameters, the result may still be sub-optimal.

Another limitation to use the  $\mu$ -Law tuning algorithm is that its applications are restricted to those systems that can be stabilized by conventional PID controllers. A future research subject is to choose and tune a proper control surface function that can be applied to more general plants.

## References

- [1] Zadeh, L. A., "Fuzzy Sets", *Information and Control*, 1965, pp 338-353.
- [2] Zadeh, L. A., "Fuzzy Algorithm", *Information and Control*, 1968, pp 94-102.
- [3] Zadeh, L. A., "Outline of A New Approach to the Analysis Complex Systems and Decision Processes", *IEEE Trans. Syst. Man. Cybern.*, Vol. SMC-3, 1973, pp 28-44.
- [4] Mamdani, E. H., "Applications of Fuzzy Algorithms for Simple Dynamic Plant", *Proc. IEE*, Vol. 121, No. 12, 1974, pp 1585-1588.
- [5] Mamdani, E. H., and Assilian, S., "An Experiment in Linguistic Synthesis with A Fuzzy Logic Controller", *Int. J. Man. Mach. Studies*, Vol. 7, No. 1, 1975, pp 1-13.
- [6] Mamdani, E. H., "Advances in the Linguistic Synthesis of Fuzzy Controller", *Int. J. Man. Mach. Studies*, Vol. 8, No. 6, 1976, pp 669-678.
- [7] Yasunobu, S., Miyamoto, S., and Ihara, H., "Fuzzy Control for Automatic Train Operation System", *Proc. 4<sup>th</sup> IFAC/IFIP/IFORS Int. Congress on Control in Transportation System*, Baden, April 1983.
- [8] Yasunobu, S., and Miyamoto, S., "Automatic Train Operation by Predictive Fuzzy Control", *Industrial Application of Fuzzy Control*, M. Sugeno, Ed. Amsterdam: North-Holland, 1985, pp 1-18.

- [9] Bare, W. H., Mulholl, R. J., and Sofer, S. S., "Design of A Self-Tuning Rule Based Controller for A Gasoline Refinery Catalytic Reformer", *IEEE Trans. On Systems, Man, and Cybernetics*, SMC-35, 1990, pp 156-164.
- [10] Liaw, C. M., and Wang, J. B., "Design and Implementation of A Fuzzy Controller for A High Performance Induction Motor Drive", *IEEE Trans. On Systems, Man, and Cybernetics*, SMC-21, 1992, pp 921-929.
- [11] Meier, R., Nieuwland, A. M., Zbinden, A. M., and Hacisalihzade, S. S., "Fuzzy Logic Controller of Blood Pressure During Anesthesia", *IEEE Control Systems*, Dec. 1992, pp 12-17.
- [12] Li, Y. F., and Lau, C. C., "Development of Fuzzy Algorithms for Servo Systems", *IEEE Control System Magazine*, Vol. 9, No. 3, April 1989, pp 65-72.
- [13] Li, H. X., "A Comparative Design and Tuning for Conventional Fuzzy Control", *IEEE Trans. On Systems, Man, and Cybernetics*, Vol. 27, No. 5, Oct. 1997, pp 884-889.
- [14] Smith, S. M., and Comer, D. J., "Automated Calibration of a Fuzzy Logic Controller Using a Cell State Space Algorithm", *IEEE Control Systems*, August 1991, pp 18-27.
- [15] Ying, H., Siler, W., and Buckley, J. J., "Fuzzy Control Theroy: A Nonlinear Case", *Automatica*, Vol. 26, 1990, pp 513-520.
- [16] Homaifar, A., and McCormick, E., "Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms", *IEEE Trans. On Fuzzy Systems*, Vol. 3, No. 2, 1995, pp 129-139.
- [17] Jang, J. R., Sun, C. T., and Mizutani, E., *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1997.
- [18] Wang, Y., *Design and Control of Spherical Gimbal for Laser Tracking System*, Ph.D. Dissertation, Florida Atlantic University, 1998.
- [19] Xu, M., *High Performance and Robust Control*, Ph.D. Dissertation, Florida Atlantic University, 1996.

## Apply a Fuzzy Logic Controller to Suppress Noises and Coupling Effects for a Laser Tracking System

Ying Bai, Zvi. S. Roth and Hanqi Zhuang

### 6.1 Introduction

Implementation of robots in industries, manufacturing and other areas demand robots to perform movements with high accuracy. Calibration is a way to improve the accuracy of the performance of robots. Collecting a high quality measurement data for robot calibration is a challenging task in the robot calibration. Utilizing a laser tracking system (LTS) is a potential solution to handle this challenging problem [1-4].

The idea of using the laser interferometer transducer system as a robot measurement technique is developed by Lau, Hocken and Haight [3]. This method is based on the principle of triangulation or point measurement technique. It involves tracking a target using a laser beam. By measuring the compound angles of the tracking motor with the distance measured by the interferometer, the position of the end-effector of the robot can be calculated. Others [5] introduced multi-laser interferometry for coordinate measuring. In such a method, 3D coordinates of the target position are computed by laser trilateration using the length data provided by three laser tracking interferometers. This method is extended further to provide position and orientation measurements by using six laser interferometry units to track three separate target points [6].

Harb [7] has reported a laser tracking system for multi-axis machine calibration. A linear tracking axis that consists of a carriage sliding on a twin recirculating motor is used. In this method, however, the calibration range is limited by the linear tracking axis, and maximum tracking speed is around 50 *mm/sec*. A 2D-plane laser tracking measurement system is reported by Nakamura *et al.* [8]. A spherical seat bearing and a steel-ball rotor are utilized in this scheme. The spherical seat bearing is set at the rotation center of the two motors. A two-dimensional calibration has been made for an industrial robot using this system with a tracking speed of about 200-300 *mm/sec* and a calibration accuracy of about  $\pm 30 \mu\text{m}$ . Rene *et al.* [9] have reported a portable laser tracking system using the triangulation principle. In their work, two laser beams are used to track a target mounted on the end-effector of the robot to obtain measurements. A measurement accuracy of about 100  $\mu\text{m}$  and a tracking speed of 570 *mm/sec*. are obtained by using the LTS to calibrate a robot.

In order to improve the tracking and processing speed, some researchers [10] developed a new method to obtain the orientation information of the target on the end-effector in real time. This method, however, needs an additional two position sensor detectors (PSDs) and more optical components, and therefore makes the system very complicated. An automatic calibration of the initial distance between the HOME position of the LTS and target is reported in [11], but again additional optical components are needed to install in the LTS.

Vincze and Spiess *et al.* [12,13] reported a laser tracking system to measure position and orientation of a robot end-effector under motion. In the relevant configuration, a floating point digital signal processor is used to speed up the processing time for the real time measurement, and a CCD imagine processing unit is utilized to obtain the orientation information of the end-effector. It is reported that the measurement accuracy of the system is around 50  $\mu\text{m}/\text{m}$  and the workspace is larger than 1 cubic meter. Some other researchers have reported a laser tracking system [14] with a similar configuration as indicated in [12]. Its tracking speed, however, is relatively slow. Spiess *et al.* [15] have addressed the calibration of a dynamic robot using a 6D laser tracking system.

In order to reduce the effects of noises, disturbances and nonlinearity on the control system of an LTS, Richard developed a two-axis fuzzy mirror controller for laser beam alignment. By comparing the fuzzy and non-fuzzy control systems used in the LTS, a significant improvement on reducing the affection of noise to the tracking error is presented. Shieh and Li [16] developed an Integrated Fuzzy Logic Controller (IFLC) for a servomotor system. With the IFLC, the performance of the original control system is greatly improved compared with a traditional PID controller. Kawaji *et al.* [17] developed a fuzzy model following a servo control system for a nonlinear DC servomotor. The experimental results showed that it could significantly reduce the effect of the nonlinearity of the DC servomotor by using the developed fuzzy model following control system. Betin *et al.* [18] developed a fuzzy logic that was applied to a speed control for a stepping motor drive to improve the tracking performance and robustness of the high stepping rates on stepping motor. Ha *et al.* [19] designed a robust control system using sliding-mode control that incorporates a fuzzy tuning technique. The fuzzy tuning schemes were employed to improve control performance and to reduce chattering in the sliding mode. To improve the control properties of a crane system for various loading conditions and varying distances, Li and Lee [20] developed and implemented a fuzzy motion control system. The experimental results showed that the position accuracy of the crane was greatly improved by using this fuzzy control strategy. Chang *et al.* [21] reported a fuzzy P+ID controller for a mechanical manipulator. Both dynamic transient and steady-state periods of the manipulator could be significantly improved under uncertainty of the system. Rubaai *et al.* [22] designed and developed a microprocessor-based fuzzy logic tracking controller for motion controls and drives. The results showed excellent tracking performance for both speed and position trajectories.

It is obvious that a challenging task of developing a laser tracking system for robots calibration is to design and implement a satisfactory controller to achieve both smooth tracking performance and high measurement accuracy. A basic requirement of the laser tracking measurement system is that during the tracking

measurement process, the outgoing laser beam from the laser transducer system and the returned laser beam from the target should have interference (overlapping) at any moment, which is equivalent to reduce the tracking error of the LTS.

In our study, two types of noises and disturbances are of main concern: one is the background and laser head (resulting from long time service of the laser tube) noise and the other is the electrical coupling effect of the motor drivers. It has been confirmed by our experimental results that the system tracking performance may greatly degrade due to this coupling effect, and it cannot be suppressed effectively by using a classical PID control strategy. Another minor issue is the nonlinearity existing in the LTS, such as the nonlinear property of the PSD.

The goal of this research is to develop strategies that can control an LTS that exhibits high precision tracking and measurement performances. The control objective is not only to minimize the target tracking error but also to suppress the coupling effects and disturbances coming from both system itself and working environment during the tracking measurement process.

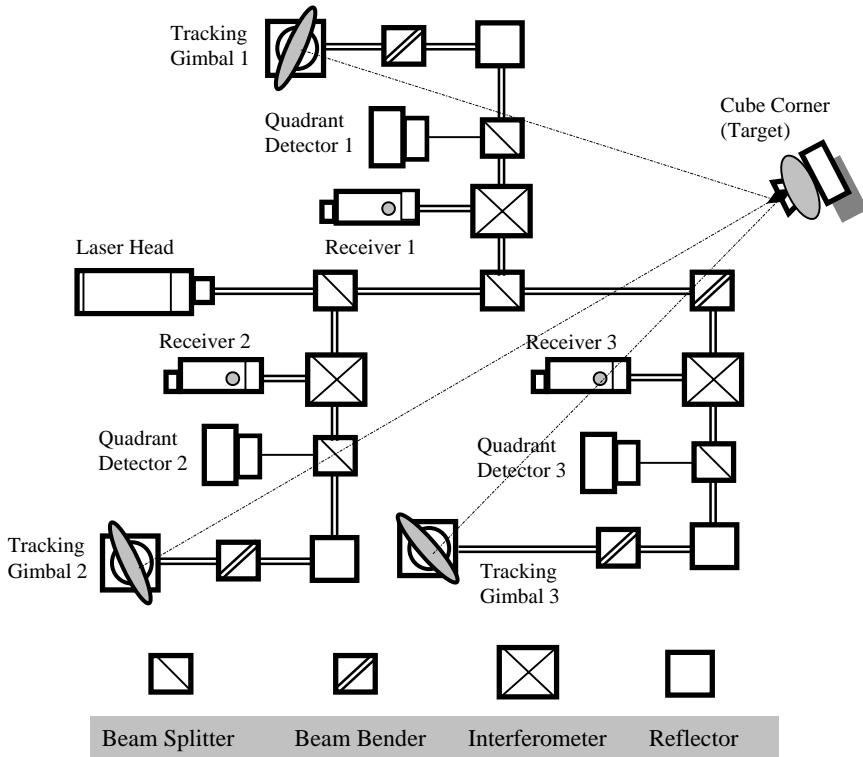
This chapter is organized in the following manner. The operating principle of the laser tracking measurement system is outlined in Section 6.2. The control system and coupling effects existed in the LTS as well as noise sources for the system are discussed and analyzed in Section 6.3. The fuzzy logic control strategy is discussed in Section 6.4. The implementation of fuzzy control system, including simulation and experimental results, are provided in Section 6.5. Section 6.6 gives the conclusion.

## 6.2 Operating Principle of LTS

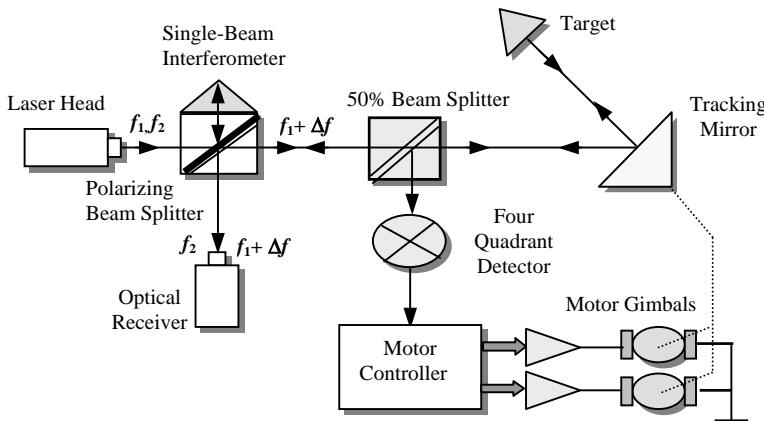
Laser tracking coordinate measurement systems are very accurate metrology devices for robots and machine tools. A laser transducer subsystem consists of one or more interferometers and related optical components, a tracking mirror mounted on two degree-of-freedom (DOF) gimbals, a corner cube retroreflector mounted on the moving target. Laser transducer subsystem consists of a quadrant photo detector (photodiode) that measures the off-center deviation of the returning laser beam from the cube corner retroreflector mounted on the target, and a signal processing unit associated. The driving subsystem consists of two DC motors, which a two-DOF gimbal is made of, motor amplifiers and feedback encoders. The digital controller consists of two controllers, inner loop standard PID controller provided by the motor manufacture, and an outer loop controller which is the main focus of this chapter.

A schematic diagram of the control mechanism of the LTS currently in use at the Robotics Center, Florida Atlantic University (FAU) is shown in Figure 6.1. This is a three-beam laser tracker system with the ability of measuring position and orientation simultaneously during the normal measurement process.

In order to have a clear picture of the operational LTS, a single-beam laser tracking system is shown in Figure 6.2.



**Figure 6.1.** Schematic process description



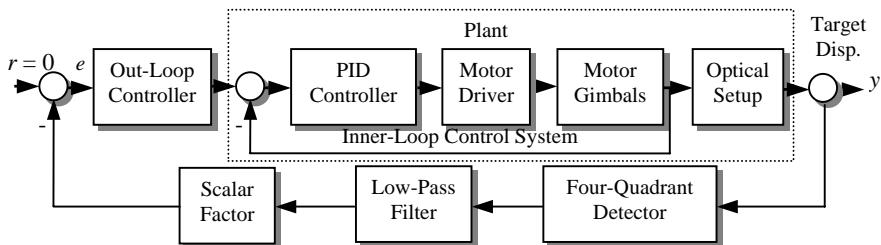
**Figure 6.2.** Schematic of a single-beam LTS

Two orthogonally polarized laser beams at narrowly spaced frequencies, the measuring beam and the reference beam, are emitted by the laser head. The reference beam is diverted to the receiver by a polarizing beam-splitter located on

the interferometer. The measuring beam proceeds through the interferometer and the 50% beam-splitter and is directed by the tracking mirror to the retroreflector located on the moving target. The returning beam from the retroreflector goes parallel to the incoming beam and enters the 50% beam-splitter from the opposite direction. The beam which passes through the 50% beam-splitter is deflected by the polarizing beam-splitter and emerges into the receiver. If the movable retroreflector changes position, a Doppler frequency shift of  $\Delta f$  occurs, which is then translated through interferometry to a relative displacement reading having a resolution in the order of magnitude of a fraction of a laser wavelength. The beam reflected by beam-splitter goes to a four-quadrant detector, which senses the deviations of the beam from the center. The sensor outputs are the error signals that drive the control systems of the two-DOF mirror gimbals.

### 6.3 The Control System, Coupling Effects and Nonlinearity

The control schematic block diagram of the laser tracking system is illustrated in Figure 6.3. It resembles the one proposed in [23] for a vision-driven robotics system. Whenever the tracking beam hits the corner of the retro-reflecting target, the returning beam from the target is coincident with the incoming beam to the target. No error signal is thus generated by the photodiode. An arbitrary target displacement causes the returning beam to go parallel to the incoming beam, in which case an error signal proportional to the target displacement is produced by the four-quadrant photodiode (Position Sensor Detector). The error signal may be preprocessed by an outer loop controller according to a differential relationship between the returning beam off-center 2D error and the gimbal position adjustments prior to being mapped to the joint space of the tracking gimbals [24]. An inner loop controller then produces proper control signals based on the desired displacement and velocity feedback of the gimbal angles.

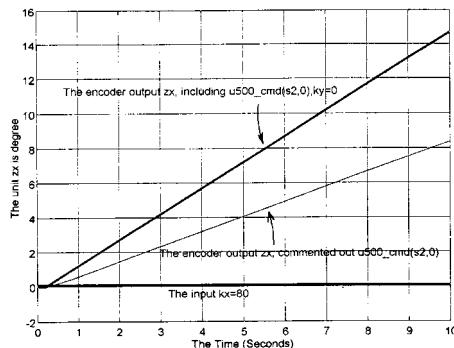


**Figure 6.3.** The control block diagram of the LTS

For the current system, the inner-loop controller is made of a commercially available motion controller. Tuning of the inner-loop PID controller is performed according to manufacturer recommended empirical procedure, as outlined in [25]. Coupling effects emerge during the tracking process of the LTS. There are two coupling effects existing in the system, one is mechanical coupling effects existing between two DC motors, and the other is electrical coupling effects existing

between different control channels in the inner loop DC motor control unit as well as the motor drivers. The inner-loop digital controller is designed assuming that the mechanical coupling effects between the two gimbals motors are negligible. This is always true in a relatively slow-tracking process. The two decoupled loops in this case can be approximated by a standard DC-motor control system. Electrical coupling effects, however, are implicitly taken into consideration in the outer-loop design.

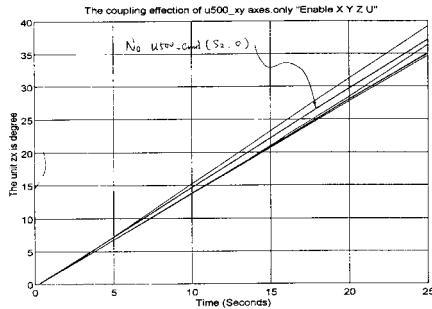
Initially a traditional PID controller was designed as an outer-loop controller for our tracking system. In the ideal case, two different control channels on the inner-loop control unit should not have any mutual interference with each other, which is the assumption for the design of the outer-loop controller. However, in practice we found that this interference is really significant for our tracking system. We called this an electrical coupling effect in our system. This coupling effect greatly degrades the tracking speed and tracking performance, and in some worse cases, the tracking system cannot work properly. We observed that the tracking performance is satisfied when one of two motors of the gimbal is driven by the controller, and another motor is kept motionless at the same time. However, the tracking performance is seriously degraded whenever two motors work simultaneously. This electrical coupling effect is shown in Figure 6.4.



**Figure 6.4.** Coupling effects between the two motors

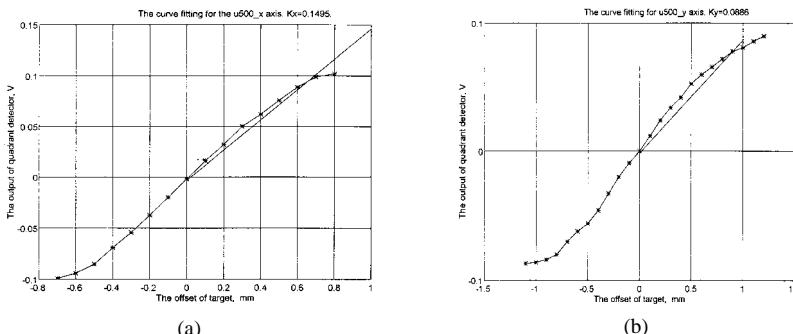
To obtain the trajectories in Figure 6.4, the outer-loop is open and the inner-loop is closed for the same input command. The gimbal consists of a big motor ( $Y$  axis) and a small motor ( $Z$  axis). The coupling effect of the big motor ( $Y$  axis) to the small motor ( $Z$  axis) is shown. The thick output is the encoder output of the small motor when both small and big motors are activated by the controller. The thin output is the encoder output of the small motor when the tracking command is only sent to the small motor by the controller. A big difference exists between the two outputs. Also this coupling effect is not a constant; it varies from time to time. This can be found from Figure 6.5. The encoder output in the middle line is the output of the small motor without sending the control command to the big motor. This means that only the small motor is working during the tracking process. The

outputs in the top and bottom of the middle line represent the output of the small motor when both small and big motors are working simultaneously. Different outputs are recorded in this figure, which means that the outputs of the small motor are different for each time even if other conditions are kept the same. It is very difficult to decouple this coupling effect because it is a variable. The pure PID controller cannot overcome this coupling effect effectively.



**Figure 6.5.** The variation of coupling effects

Another problem we encountered in our outer-loop controller design is some nonlinear effects in the system, which cannot be modeled explicitly and accurately. One of them is the photodiode sensor characteristic, which is shown in Figure 6.6a, (vertical motion) and Figure 6.6b, (horizontal motion), respectively. The units of input/output for the sensors are offset ( $mm$ )/voltage (V). It is obvious that the nonlinearity of the sensor in the horizontal direction is greater than that of the sensor in the vertical direction.



**Figure 6.6.** Sensor characteristic for vertical (a) and horizontal (b) motions

Based on the above analysis of the coupling effects and nonlinearity existing in our system, a fuzzy logic controller will be designed as an outer-loop controller in our tracking system. It has been reported that the fuzzy logic controller is a robust controller and it has a better performance for nonlinear systems control [26-28].

This has been confirmed by the simulation and experimental results of LTS at Robotics Center at FAU.

## 6.4 Design of Fuzzy Logic Controller

We employed different control strategies for the two different gimbal motors. For the vertical tracker, a fuzzy controller is designed which includes both coarse (fast tracking speed) and fine (small steady state error) fuzzy control strategies. For the horizontal tracker, a mixed PID and fine fuzzy controller is designed because a bigger DC motor is used in the horizontal tracker and it turns out to be more difficult to tune with a pure fuzzy controller due to a large moment of inertia of the motor.

The design of the FLC followed the standard steps of defining input and output membership functions (MFs), constructing a group of control rules, and mapping these rules to a lookup table that produces final defuzzification outputs. A gain factor  $G$  is added to the output of the lookup table for the tuning of the fuzzy control output, and a bilinear interpolation algorithm is also added to the output of the lookup table to improve the smoothness of the output of the FLC. The block diagram of FLC is shown in Figure 6.7.

It is found empirically that truncated Gaussian shaped membership functions yielded smoother tracking performance and prompt time response in comparison with triangular or trapezoidal functions. To create a set of linguistic variables, seven Gaussian MFs are used to represent each of the three variables: input error, error rate and control. These MFs are defined as: Large Negative (LN), Medium Negative (MN), Small Negative (SN), Zero (ZE), Small Positive (SP), Medium Positive (MP), and Large Positive (LP).

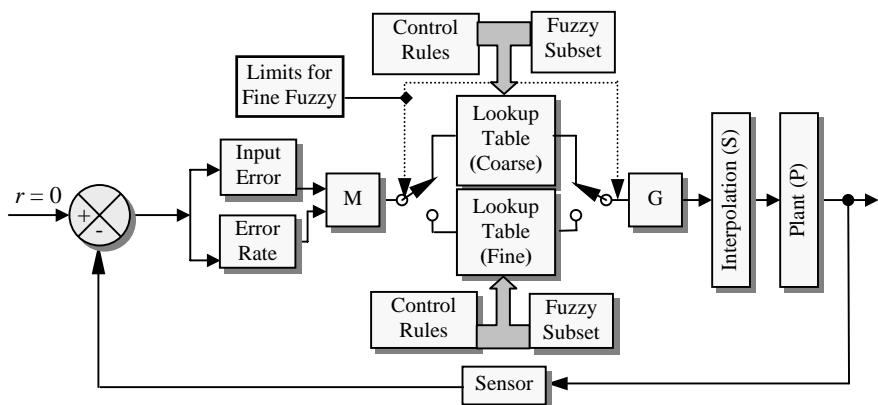


Figure 6.7. Block diagram of the FLC

The ‘Small’ membership functions for both position and control variables are chosen to yield zero value at Zero, which is helpful in suppressing excessive overshoots and limit cycles.

For the vertical gimbal axis, a set of rules is selected based on experimentation. For each error ( $e$ ) and error rate ( $e'$ ) pair, a control variable can be chosen from the table shown in Figure 6.8. A total of 49 rules are constructed for the vertical tracking gimbal axis control.

$e \backslash e'$	LN	MN	SN	ZE	SP	MP	LP
LP	ZE	SP	SP	MP	MP	LP	LP
MP	SN	ZE	SP	SP	MP	LP	LP
SP	MN	SN	ZE	SP	SP	LP	LP
ZE	LN	MN	SN	ZE	SP	MP	LP
SN	LN	LN	SN	SN	ZE	SP	MP
MN	LN	LN	MN	SN	SN	ZE	SP
LN	LN	LN	MN	MN	SN	SN	ZE

Figure 6.8. The control rules table

A mapping relationship exists between a step response in the time domain and an error to error-rate trajectory in the phase plane, which is shown in Figures 6.9a and 6.9b, respectively [29].

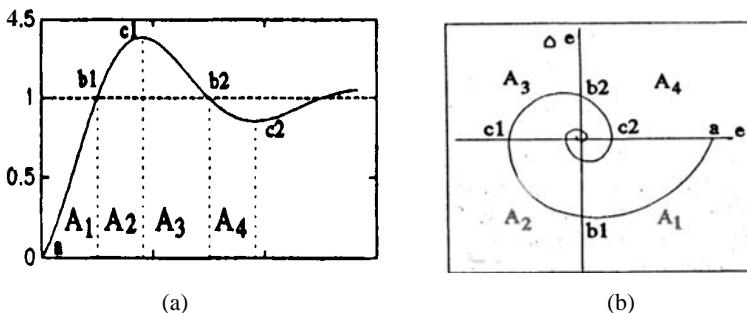


Figure 6.9. Mapping between step response and error to error-rate trajectory: (a) step response in time domain; (b) error and error-rate trajectory in phase plan

It can be found that the step response in the time domain can be divided into four areas,  $A_1$  to  $A_4$ , peak and valley points,  $c_1$  and  $c_2$ , as well as the crossing-over points,  $b_1$  and  $b_2$ . For an actual system, a time delay exists in the dynamic model, and this delay can be represented as a shift to the right with amount of delay period of the time in the step response (6.9a). If this delay is mapped to the phase plan, it is equivalent to shift  $A_1$  and  $A_2$  to the left, and shift  $A_3$  and  $A_4$  to the right. We can map these shifts to the control rule table we developed above in Figure 6.8, and shift the bottom rules to the right, and shift the top rules to the left. This shift is a

relative movement and it is equivalent to shift the trajectory in  $A_1$  and  $A_2$  to the left,  $A_3$  and  $A_4$  to the right, which is shown as two arrows in Figure 6.8. By employing this shifted control rule table, we can effectively improve the transient response to a system that has a long time delay. This has been confirmed by our experimental results.

Adjustment and rotation of the quadrant detector with respect to an axis defined by the incoming laser beam allows a good decoupling of the two sensor output signals. That is, errors in the vertical ( $x$ ) and horizontal ( $y$ ) directions of the four-quadrant detector are sent respectively to each of the axes of the gimbals. But this adjustment is very limited in our actual system because the photodiode sensor units (including the opto-electronic signal translation, signal amplifying and low-pass filter) are fixed on the optical platform in our tracking measurement system. We found that this coupling effect can be negligible if the four quadrant detectors are well aligned optically by using the optics alignment method, that is, the output of the quadrant detector is zero when the incident laser beam hits the center of the quadrant detectors. The inputs of each outer-loop controller are the error and error rate of the target motion projected onto the quadrant detector in either the  $x$  direction or the  $y$  direction. The output is the required angular velocity of the respective gimbal axis that will drive both the error and error rate of the target to zero. The error rate is computed from the errors through a simple finite difference.

Once the lookup table is constructed, no further modification of its structure or entries is ever attempted. The design parameters for the actual experimental tuning process are the input variables scale factors and the output control gains. The tuning of the scalar factors is first performed on a simulated model of the closed-loop system. For that purpose, the model of each motor and its drive system is actually identified by using the MATLAB® Identification Toolbox.

The horizontal axis is controlled by a mixed-fuzzy controller. The PID control is utilized when the tracking error is relatively greater in order to obtain a relative faster tracking speed, and a fine fuzzy control is used to obtain a small static state error and therefore a high tracking accuracy. There is a function used to switch from PID control to fine fuzzy control when the input tracking error is within the range of the pre-defined ‘small’ error.

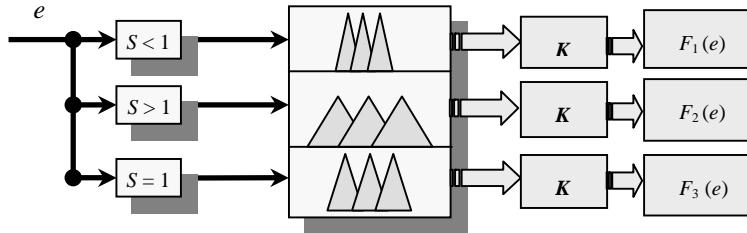
During the tuning of the FLC, it is necessary to use many different membership functions (MF). H. Zhuang and X. Wu reported a technique to modify the membership function to obtain the optimal performance for an LTS [30]. Scale factors can be used to obtain multiple MFs from only one MF. An input error membership function is shown in Figure 6.10. A scale factor,  $S$ , is used to multiply the input error to obtain different ranges of the input error. As shown in Figure 6.10, by changing the scale factor  $S$ , different MFs can be obtained:

When  $S=1$ , original membership function is obtained

When  $S>1$ , a more coarse membership function is obtained

When  $S<1$ , a finer membership function is obtained

One can use the same method to obtain different membership functions for error rate.



**Figure 6.10.** The tuning of membership functions

With turning of the membership functions, during the tuning of an FLC, many different lookup tables should also be tested in order to achieve the optimal performance for the control system. It is time consuming to construct a new lookup table based on the tuned membership function and control rules each time during the tuning process. An easier method is to use a control gain factor,  $K$ . The output of the lookup table can be conveniently changed and tuned by adjusting of this control gain factor. The principle of the tuning control gain factor is:

- When  $K < 1$ , a finer lookup table is obtained
- When  $K > 1$ , a coarser lookup table is obtained
- When  $K = 1$ , the original lookup table is obtained

This tuning is only to enlarge or contract the lookup table uniformly, and therefore to modify the associated output element in the lookup table.

## 6.5 Simulation and Experimental Results

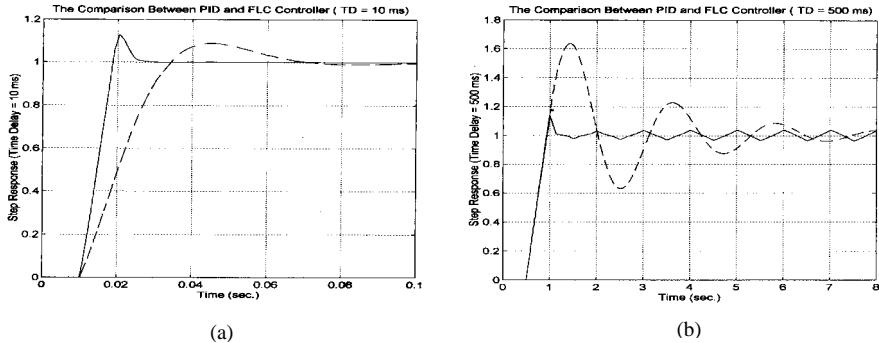
An identified dynamic system model (including the inner-loop controller and motors as well as motor drivers) for the  $x$ -axis is given below [31]:

$$G(s) = 6519 \frac{10.68s + 6519}{s(s + 430.6)} e^{-0.005s}$$

The pure time delay of 5 ms is attributed to the communication time between the computer controller and the motor drivers. The control system for both PID and FLC based on the dynamic model given above is simulated by using MATLAB® Simulink® Toolbox.

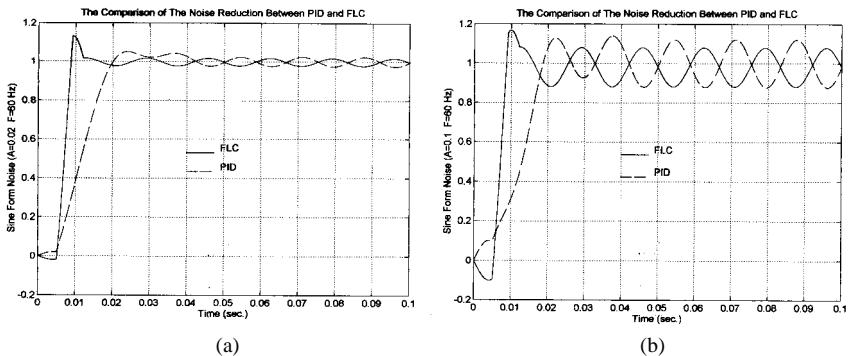
For an ideal LTI control system with no time delay, no noise and disturbance, the control performances are similar either using a PID controller or an FLC.

The simulation study is then divided into three categories: time delay, noise and disturbance, and uncertain system models. Figure 6.11 shows the simulated results for the time delays of 10 ms (a) and 500 ms (b), respectively.



**Figure 6.11.** Simulated results for PID and FLC with time delays: (a) 10 ms; (b) 500 ms

The dashed line represents the PID control performance and the solid line represents the FLC performance. It can be found that the settling time of the FLC is much shorter compared with that of the PID controller when the time delay is 10 ms, and the PID controller even has an unacceptable overshoot as the time delay is extended to 500 ms for the given system. It is also noticed that the FLC has a relatively large steady state error as the time delay approaches to 500 ms. In order to avoid this steady state error, a fine fuzzy logic controller can be utilized to reduce this error. For the simulation purpose, however, we only use a coarse fuzzy logic controller to compare with the PID controller.



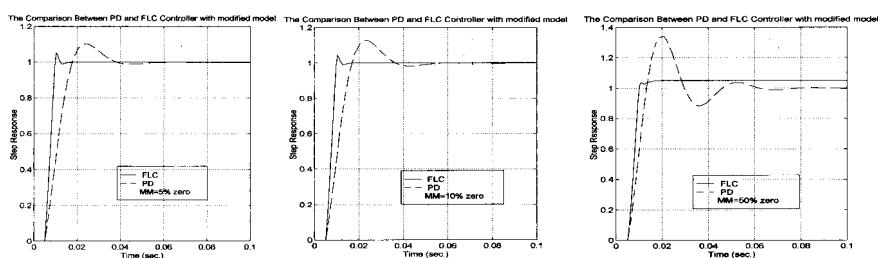
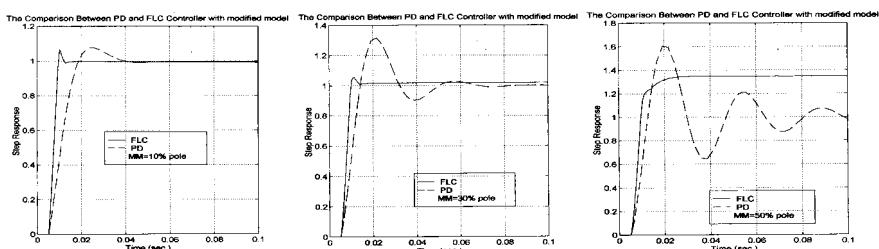
**Figure 6.12.** Simulated results for: (a) PID; (b) FLC, both with noise affection

To simulate the noise effect to the PID and FLC system, a 60-Hz sinusoidal signal is utilized. Figure 6.12 shows the simulated results for a variation of magnitudes of sinusoidal waveform, from 0.02 (a) to 0.10 (b), which is equivalent to 2% and 10% compared with a unit step signal. It can be found that the FLC is about 30% better in reducing the noise effect.

We simulate an uncertain system with a variation of zeros and poles in different rates, which is shown in Table 6.1, and the control performances of PID and FLC for these uncertain systems are shown in Figures 6.13 (uncertain zeros) and 6.14 (uncertain poles), respectively.

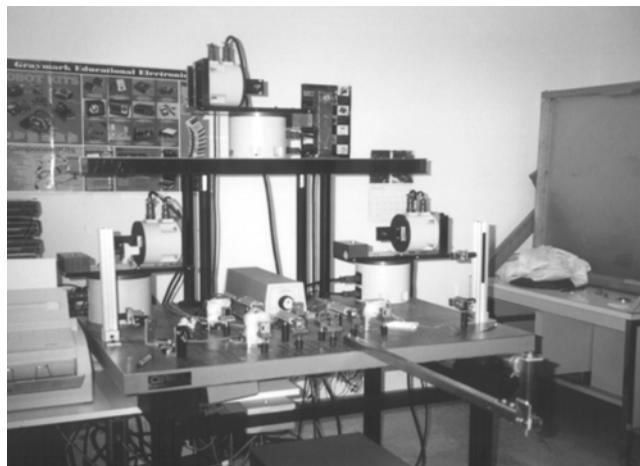
**Table 6.1.** Simulated zeros and poles for uncertain system models

Zero/Pole	Change rate	Modified system model
Zero	5%	$10.68 \frac{s + 641.0}{s(s + 430.6)} e^{-0.005s}$
Zero	10%	$10.68 \frac{s + 671.5}{s(s + 430.6)} e^{-0.005s}$
Zero	50%	$10.68 \frac{s + 915.8}{s(s + 430.6)} e^{-0.005s}$
Pole	10%	$10.68 \frac{s + 610.5}{s(s + 473.7)} e^{-0.005s}$
Pole	30%	$10.68 \frac{s + 610.5}{s(s + 301.4)} e^{-0.005s}$
Pole	50%	$10.68 \frac{s + 610.5}{s(s + 215.3)} e^{-0.005s}$

**Figure 6.13.** Comparison between PID control and FLC with uncertain zeros**Figure 6.14.** Comparison between PID control and FLC with uncertain poles

It can be found that the FLC has better control performance compared with the PID controller for both uncertain zeros and poles. The uncertain poles have a larger effect on the system performance than that of zeros. The PID control performance has an unacceptable overshoot as the uncertain pole approaches to 50%. Also the settling time of the PID control system is much longer compared with the FLC system as the uncertainty of zeros or poles becomes larger and larger. The shortcoming of the FLC for the uncertain system model is that the steady state error will become larger as the uncertain rate of zeros and poles approaches 50%. This can be compensated with a fine FLC in the actual application, and this steady state error can be controlled into an acceptable range.

For experimental verifications, a target retroreflector is mounted on either a small MITUTOYO coordinate measuring machine (CMM), or on the end-effector of a PUMA 560 robot. Figure 6.15 shows the experimental setup of LTS in FAU Robotics Center. Three trackers are installed at two optical platforms, and all optical components, including the laser head, are installed on the base platform (bottom platform). The metal bar is used to mount the retroreflector at the beginning of the tracking measurement and this can be considered as the home position or reference point of the LTS. The four-quadrant detector's signals are digitized into a Pentium PC. The projected target error values are then numerically differentiated to obtain error rates, and are jointly fed to the fuzzy logic controller. Using error and error rate at each sampling interval as the indices, the position control command is computed by interpolating 4 neighboring control values fetched from the lookup table. This control command, scaled by a control gain, is then sent to a motion controller to drive individual gimbal motor.



**Figure 6.15.** Experimental setup of the LTS

Currently, the sampling period of the outer-loop control is 2 ms for a single gimbal system and 6 ms for a three-gimbal system (inner-loop sampling period is

0.25 ms). Target maximum tracking speed is about 60 mm/sec., with the target being 2 meters away from the laser trackers.

In order to verify that the FLC has a better ability to reduce the noise and disturbance for our LTS, a noise spectrum analysis for the small motor gimbal is performed. Figure 6.16 shows this spectrum analysis for the PID controller (a) and FLC (b).

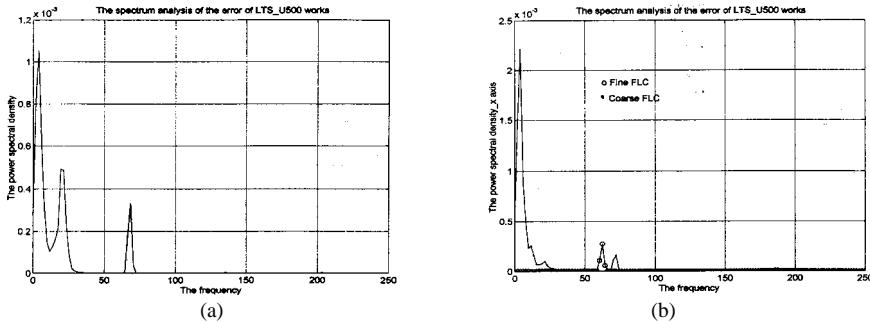


Figure 6.16. Spectrum analysis of PID and FLC for small motor gimbal

In Figure 6.16a, the noise with low frequency (0~30 Hz) is coming from the background and laser head, and the noise or disturbance with a frequency of 60 Hz is coming from the electromagnetic components in the motor driver, which is called the coupling effect in the current study. When applying a PID controller, the noises and disturbances have not been significantly reduced. In Figure 6.16b, the solid line represents the spectrum of a coarse FLC and the circle represents a fine FLC. It can be found that the second peak of low-frequency noises has been removed by applying a coarse FLC, but the coupling effect is still in there. By applying a fine FLC, the low-frequency noise is disappeared and coupling effect is also reduced.

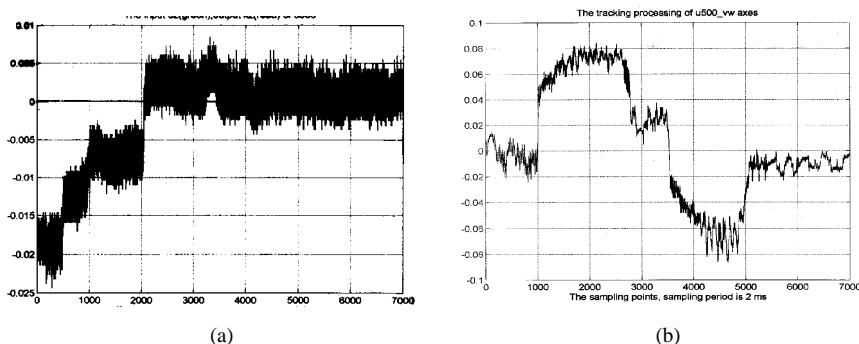


Figure 6.17. Steady state error with: (a) FLC; (b) PID controller

Figure 6.17 shows a dynamic tracking trajectory by using the FLC (a) and PID controller (b), respectively. In Figure 6.17a, the first section illustrates the tracking errors when the target moves from the negative to the positive directions, and finally stops around the balance position. The tail section shows the range of the steady state error of the FLC (peak-to-peak is about 0.005 V). Compared with the steady state errors in the PID controller in Figure 6.17b, which is about 0.02 V (peak-to-peak value), a ratio of 4:1 in error magnitude is obtained between the PID controller and FLC.

## 6.6 Conclusion

The design and implementation of a fuzzy logic controller for the laser tracking system is discussed and presented in this chapter. The effectiveness of the FLC in suppression of noises and disturbances is assessed with both simulation and experimental studies, and the comparison between the PID control and FLC for both transient and steady state performance is also conducted based on the experimental results obtained from Robotics Center at Florida Atlantic University. Evidently the FLC has a superior performance than that of PID control, especially in the environment in which relatively large disturbances and noises are presented for the control systems.

## Acknowledgement

This work is supported in part by a National Science Foundation grant.

## References

- [1] K. Lau, Dagalakis and D. Myers, "Testing", International Encyclopedia of Robotics, *Application and Automation*, R. C. Dorf and S. Y. Nof, Willey, 1988, pp 1753-1769.
- [2] K. Lau, R. Hocken and L. Haynes, "Robot Performance Measurements Using Automatic Laser Tracking Techniques", *Robotics and Computer-Integrated Manufacturing*, Vol.2, No.3, 1985. pp 227-236
- [3] K. Lau, R. Hocken and W. Haight, "An Automatic Laser Tracking Interferometer System for Robot Metrology", National Bureau of Standards, Gaithersburg, MD, March, 1985. pp 113 – 152.
- [4] L.B. Brown, J.B. Merry and D.N. Wells, "Coordinate Measurement with a Tracking Laser Interferometer", *Laser & Applications*, Oct., 1986, pp 69-71.
- [5] *Laser Trilateration – a Fast, Accurate and Flexible Measuring Technique*, Quality Today, May 1989. pp 87 – 121.
- [6] Slocum, "Development of Six-Degree of Freedom Position and Orientation Sensing Device: Design, Theory and Testing", *Intl. Journal of M/C and Manufacture*, Vol. 28/4. 1988. pp 225 – 240.
- [7] S.M. Harb, "Laser Tracking System for Multi-Axis Machine Calibration", *International Conference on Laser Metrology and Machine Performance*, Southampton, England, 2nd, 1995, pp 161-171.

- [8] O. Nakamura *et al.*, “A Laser Tracking Robot-Performance Calibration System Using Ball-Seated Bearing Mechanisms and a Spherically Shaped Cat’s-Eye Retroreflector, *Rev. Sci. Instrum.* 65(4), April 1994, pp 1006-1011.
- [9] J.R. Rene, “A Portable Instrument for 3-D Dynamic Robot Measurements Using Triangulation and Laser Tracking”, *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 4, August 1994, pp 504-516.
- [10] Y. Bao and N. Fujiwara, “Dynamic Measurement Orientation by LTS”, *Proceedings of the Japan-USA Symposium on Flexible Automation*, 1996, pp 545-548.
- [11] Y. Bao and N. Fujiwara, “Automatic Initial Distance Measurement Without Angles for Laser Tracking System”, *Proceedings of the Japan-USA Symposium on Flexible Automation*, 1996, pp 545-548.
- [12] M. Vincze, J.P. Prenninger and H. Gander, “A Laser Tracking System to Measure Position and Orientation of Robot End-Effectors Under Motion”, *The International Journal of Robotics Research*, Vol. 13, No. 4, August 1994, pp 305-314.
- [13] S. Spiess, M. Vincze and M. Ayromlou, “On the Calibration of a 6D Laser Tracking System for Contactless Dynamic Robot Measurements”. *IEEE Instrumentation and Measurement Technology Conference, Ottawa, Canada*, May 19-21 1997, pp 1203-1208.
- [14] C.J. Leigh-Lancaster, “Development of a Laser Tracking System”, *Proceedings of Fourth Annual Conference on Mechatronics and Machine Vision in Practice, Toowoomba, Australia*, September 23-25, 1997, pp 163-168.
- [15] S. Spiess, M. Vincze and M. Ayromlou, “On the Calibration of a 6-D Laser Tracking System for Dynamic Robot Measurements”, *IEEE Transactions on Instrumentation and Measurement*, Vol. 47, No.1, February 1998, pp 270-274.
- [16] M.Y. Shieh and T.H.S. Li, “Implementation of Integrated Fuzzy Logic Controller for Servomotor System”, *IEEE International Conference On Fuzzy Systems*, 1995, pp 1755-1762.
- [17] S. Kawaji and N. Matsunaga, “Design of Fuzzy Model Following Servo Control Systems”, *IEEE International Conference On Fuzzy Systems*, 1995, pp 1763-1768.
- [18] F. Betin, D. Pinchon and G.A. Capolino, “Fuzzy Logic Applied to Speed Control of a Stepping Motor Drive”, *IEEE Transaction on Industrial Electronics*, Vol. 47, No. 3, June 2000. pp 610 – 621.
- [19] Q.P. Ha, Q.H. Nguyen, D.C. Rye and H.F. Durrant-Whyte, “Fuzzy Sliding-Mode Controllers with Applications”, *IEEE Transaction on Industrial Electronics*, Vol. 48, No. 1, February 2001. pp 38 – 45.
- [20] C. Li and C.-Y. Lee, “Fuzzy Motion Control of an Auto-Warehousing Crane System”, *IEEE Transaction on Industrial Electronics*, Vol. 48, No. 5, October 2001. pp 983 – 993.
- [21] W. Li, X.G. Chang, J. Farrel and F.M. Wahl, “Design of an Enhanced Hybrid Fuzzy P+ID Controller for a Mechanical Manipulator”, *IEEE Transactions on System, Man, and Cybernetics – Part B: Cybernetics*, Vol. 31, No. 6, December 2001. pp 938 – 945.
- [22] A. Rubaai, D. Ricketts and M.D. Kankam, “Laboratory Implementation of a Microprocessor-Based Fuzzy Logic Tracking Controller for Motion Control and Devices”, *IEEE Transaction on Industrial Applications*, Vol. 38, No. 2, March/April 2002. pp 448 – 455.
- [23] N. Papanikolopoulos, P.K. Khosla, and T. Kanade, “Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision”, *IEEE Trans. Robot. Automat.*, Vol.9, No.1, Feb. 1993, pp 14-35.
- [24] Z.S. Roth, H. Zhuang, and S. Hu, “A Control Strategy for a Laser Tracking Measuring System”, Proc. 7th Conf. on Recent Advances in Robotics, 1994, Gainesville, FL.
- [25] Aerotech Inc., “Manual Tuning for Motors Without Tachometers on the Unidex 500”, 1994. pp 85 – 100.

- [26] Hao Ying, W. Silver and J.J. Buckley, "Fuzzy Control Theory: A Nonlinear Case", *Automatica*, 26, 1990. pp 513-520.
- [27] G.C.H Wang, S.C. Lin, "A Stability Approach to Fuzzy Control Design for Nonlinear Systems", *International Journals for Fuzzy Sets and Systems*, Vol. 48, 1992. pp 279-287.
- [28] S.B Overie, B. Demaya, J.M. Le Quellec, and A. Titli, "Performance Evaluation of Fuzzy Control Through an International Benchmark", *IFSA 93*, Seoul Korea, July 1993, pp 220 – 246.
- [29] H.X. Li and H.B. Gatland, "A New Methodology for Designing a Fuzzy Logic Controller", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 3, March 1995, pp 505 – 512.
- [30] H. Zhuang and X. Wu, "Membership Function Modification of Fuzzy Logic Controllers with Histogram Equalization", *IEEE Trans. On System, Man, and Cybernetics – Part B: Cybernetics*, Vol. 31, No. 1, February 2001. pp 125 – 131.
- [31] Y. Bai, Z.S. Roth and H. Zhuang, "Control Considerations in the Design of A Laser Tracking CMM", Proc. of the 1997 Florida Conf. on Recent Advances in Robotics, April 10-11, 1997, Miami, Florida.

## **Fuzzy Logic for Image Processing: Definition and Applications of a Fuzzy Image Processing Scheme**

Mario I. Chacón M

### **7.1 Introduction**

Image processing is a prominent area that supports applications in different fields, such as medicine, astronomy, national security, autonomous systems, product quality, industrial applications, *etc.* [1]. A high volume of research work is published every year in this area. These publications cover different topics in the image processing area. One of the most challenge works is the development of image processing algorithms to try to simulate in some way the human visual system. As is known, these kinds of algorithms require modeling of complex systems, which require processing of information with high degree of uncertainty and subjectivity. Some algorithms developed under the scheme of traditional logic, crisp approach, have been developed [1]. As a consequence, information uncertainty and subjectivity are difficult to model in the algorithms. A better structure to model a real system, in this case, the human visual system, must incorporate tools to model better subjective concepts and must be able to deal with the uncertainty of the information. A mathematical framework that incorporates these tools is fuzzy logic. Fuzzy logic allows one to model uncertainty and subject concepts in a better form than crisp models [2-10].

In the area of image processing, especially in industrial applications, it is common to deal with subjective concepts like brightness, edges, uniformity, measurements, *etc.* For example, in cosmetic inspection of products, product quality is determined based on subjective observations of the inspector on a product. Therefore, a vision inspection system used in this type of application needs to manage subjectivity and uncertainty. The concepts related to image analysis contain a certain amount of uncertainty. It is well known for example that the definition of border pixels is a hard task. The decision whether a pixel belongs to the background or to the object is non-trivial on occasion due to the uncertainty present on object edges. The results of “crisp” based algorithms in image analysis tasks are in some cases satisfactory; however if more robust systems are required, development of new algorithms that involve fuzzy logic are required. This is why incorporation of fuzzy logic into the development of image processing and analysis has opened a new research area in the image-processing field.

This chapter attempts to formulate a mathematical model for a fuzzy image processing approach to provide a guidance to perform fuzzy image processing, and illustrates the proposed methodology with some applications.

## 7.2 Fuzzy Image Processing Scheme Definition

Before we can apply fuzzy processing to an image, it is necessary to map the original image into the fuzzy domain. Since in some cases we are trying to represent a concept or characteristic of the image related to the perception we may call the new domain the *perception* domain [3]. The process of the proposed fuzzy image processing scheme Figure 7.1 is then to map the original image into the fuzzy or *perception* domain, apply a fuzzy operator to the fuzzy image, and defuzzification of the fuzzy image to return to the original domain.

The formal fuzzy image-processing scheme defined in Equations (7.1) to (7.3) is as follows.  $I(x, y)$  is the original image,  $F$  defines a set of mapping functions to map the original image into the fuzzy domain, (0,1), or *perception* domain. The *perception* domain is used when the image processing algorithm deals with the *perception* of a subjective characteristic of the image. The *perception* mapping is also made to the (0,1) interval. However, the qualifier of *perception* provides a better meaning in some cases, as it will be illustrated in other sections.

The image in the fuzzy domain is represented by  $I_F(x, y)$ . The fuzzy image is processed by a fuzzy set of operators  $\mathfrak{I}$  yielding a new fuzzy image  $I_F'(x, y)$ . Finally, the fuzzy image is mapped back to the original domain through the defuzzification function  $D$ . The processed image in the original domain is represented by  $I(x, y)'$ :

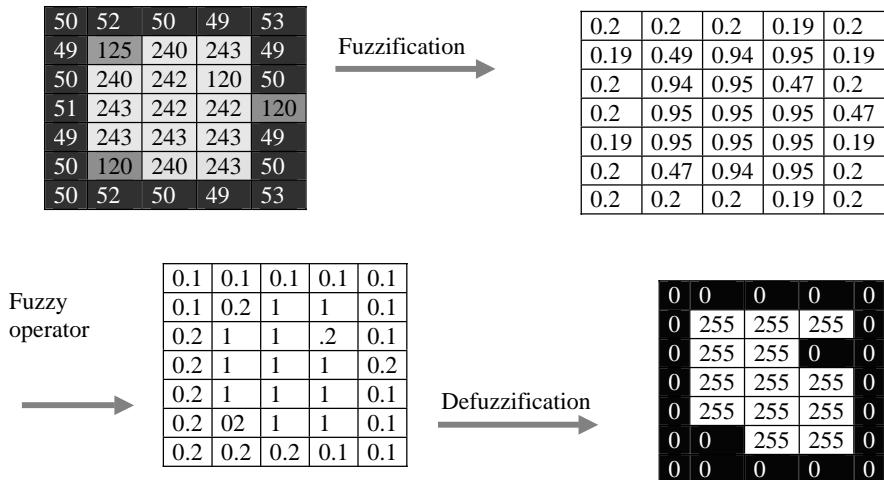
$$F : I(x, y) \Rightarrow I_F(x, y) \quad (7.1)$$

$$\mathfrak{I}[I_F(x, y)] = I_F'(x, y) \quad (7.2)$$

$$D : I_F'(x, y) \Rightarrow I'(x, y) \quad (7.3)$$

### 7.2.1 Mapping the Image into the Fuzzify Domain

The mapping function  $F$  is defined such that the image characteristics or concepts that are of interest, brightness, contrast, edges, regions, connectivity, image *complexity*, etc., could be better represented in the new image model. These characteristics or concepts are then represented by *linguistic variables* that better describe them [2]. For example, if we are interested in representing the concept *average brightness* of an image we can use the set of *linguistic variables*  $\{low, medium, high\}$  that could be defined by their corresponding fuzzy set, Figure

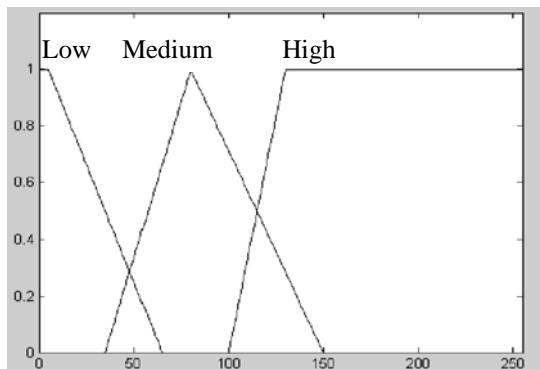
**Figure 7.1.** Fuzzy image processing scheme

7.2. In order to improve our representation of the concept we can also use hedges, fuzzy set shape transformers. Hedges can be considered as adjectives in a human language. Some examples of these modifiers are, *very*, *not very*, *about*, *etc.*

Some examples of  $F$  are the following. Definition of *brightness* of pixel  $l$  may be [11]:

$$F(l) = ml + b \quad (7.4)$$

where  $l \in L$ ,  $L$  is the set of gray level values of a pixel.

**Figure 7.2.** Representation of *average brightness* of an image

The definition of *edge* in an image could be [12].

$$F(\nabla) = S(\nabla; a, b, c) = \begin{cases} 0 & \nabla \leq a \\ 2\left(\frac{\nabla - a}{c - a}\right)^2 & a \leq \nabla \leq b \\ 1 - 2\left(\frac{\nabla - c}{c - a}\right)^2 & b \leq \nabla \leq c \\ 1 & \nabla \geq c \end{cases} \quad (7.5)$$

where  $\nabla$  stands for the gradient value of a neighborhood around a pixel.

Image *complexity* may be another concept to represent, assuming image *complexity* as the amount of attention required by a person to analyze an image. The complexity of an image,  $\Psi$ , can be mapped for example through a clustering function over the information of several images [13]:

$$FCM : \{ \psi[I_{v1}(x, y), \psi[I_{v2}(x, y)], \dots, \psi[I_{vn}(x, y)]\} \rightarrow \psi_i \quad (7.6)$$

for  $i = 1, 2$

where  $I_{vn}(x, y)$  represents an image with pixels values obtained from a gradient operator, and  $\Psi$  defines the complexity of the image.

The mapping function  $F$  may be modified to represent better characteristics or concepts by using hedges. For example,  $\alpha F(l) = \alpha(ml + b)$  may represent *very bright*, where  $\alpha$  is a hedge.

### 7.2.2 Operation on the Fuzzy Domain

The fuzzy operators  $\mathfrak{I}$  can be expressed through mathematical expressions or fuzzy rules. Some fuzzy operator examples are the following.

The contrast intensification [3] operator is a well-known fuzzy operator. The effect of the contrast intensification on fuzzy data is to reduce the uncertainty present in the data. Remember that the maximum uncertainty corresponds to membership values of 0.5. The contrast intensification operator increases the membership of values equal or greater than 0.5 and decreases the values less than 0.5. If the data represent gray level values, the effect of this operator will increase the contrast of an image. The contrast intensifier operator is defined by

$$INT(A) = \begin{cases} 2\mu_A^2(l) & 0 \leq \mu_A(l) < 0.5 \\ 1 - 2(1 - \mu_A^2(l)) & 0.5 \leq \mu_A(l) \leq 1 \end{cases} \quad (7.7)$$

where  $\mu_A(l)$  is the membership of the gray level  $l$  to the fuzzy set  $A$ .

Filtering is a common operation on images. This operation allows us to emphasize important information or discard it. For example, the output of a fuzzy filter may be defined by [14]

$$y(i, j) = \frac{\sum_{(r,s) \in A} F[x(i+r, j+s)] \square x(i+r, j+s)}{\sum_{(r,s) \in A} F[x(i+r, j+s)]} \quad (7.8)$$

where  $x(i, j)$  is the input of the 2D filter,  $F[x(i, j)]$  is a general window function that defines the type of filter, and  $A$  is the order of the window. Common filters used in image processing, the median filter, and the moving average filter are defined by the general window functions [14]

(7.9)

$$F_{\text{median}}[x(i+r, j+s)] = \begin{cases} 1 & \text{for } x(i+r, j+s) = x_{\text{median}}(i, j) \\ 0 & \text{otherwise} \end{cases}$$

$$F_{\text{mavg}}[x(i+r, j+s)] = 1 \text{ for } r, s \in A \quad (7.10)$$

If we want to have a measure of information content, we can use the fuzzy entropy [15]

$$E_{(A)} = - \sum_{i=1}^N \mu_A(l_i) P(l_i) \log P(l_i) \quad (7.11)$$

The information generated by the fuzzy operators can be used to take decisions by means of fuzzy rules of the form

*Rule k: If Fuzzy Condition Then Fuzzy Action-k*

### 7.2.3 Defuzzification Function

With respect to the defuzzification function we can find two cases. In the first case the fuzzy operators generate useful information ready for visualization. In this case the defuzzification function consists of the mapping of fuzzy values to allowed values of the visualization device, for example

$$D : (0,1) \Rightarrow (0, L-1) \quad (7.12)$$

Here, it is assumed that the visualization device operates on the range  $(0, L-1)$ . In other cases, the fuzzy values need to be defuzzyfied in order to obtain useful information, for example

$$D : D' \{ \mu_A(x) \} \Rightarrow I(x, y)' \quad (7.13)$$

A complete description of specific defuzzification methods like max member principle, centroid method, weighted average method, and mean max membership can be consulted in [16].

## 7.3 Examples of Fuzzy Image Processing

In this section several fuzzy image processing applications are described to illustrate the previous fuzzy image processing scheme.

### 7.3.1 Fuzzy Binarization

Image binarization consists of the mapping of the image pixels into black pixels and white pixels. Binarization of an image can be stated as a classification task. In this problem, we can consider two classes of pixels, background pixels and object pixels. The solution consists of the classification of the pixels into the two classes minimizing the classification error. Because class distributions are not completely disjointed in general, the pixels in the overlap interval are difficult to classify. These pixels have a high degree of uncertainty. A fuzzy clustering algorithm is used in the classification to manage the uncertainty of these pixels [11]. The benefit of this process is that the fuzzy scheme incorporates more information than the crisp algorithms.

Assume a gray scale image composed by an object on a background. The fuzzification function used to map the pixels values  $l_j$  of the image into the fuzzy domain is

$$\mu_{1j}(l_j) = (\text{gray level of } l_j) / 255 \quad (7.14)$$

and

$$\mu_{2j}(l_j) = 1 - \mu_{1j}(l_j) \quad (7.15)$$

where  $\mu_{1j}(l_j)$  represents the membership of a pixel to the object and  $\mu_{2j}(l_j)$  the membership of a pixel to the background.

The fuzzy function used to make the classification was the fuzzy clustering algorithm [17]. The computation of the fuzzy centroids,  $V_i$ , that represent the two classes is

$$V_i = \sum_{j=1}^n (\mu_{ij})^m X_j \Bigg/ \sum_{j=1}^n (\mu_{ij})^m \quad (7.16)$$

The membership updates of the pixels are accomplished by

$$\mu_{ij} = \left( \frac{1}{d^2(X_j, V_i)} \right)^{\frac{1}{m-1}} \left/ \sum_{i=1}^c \left( \frac{1}{d^2(X_j, V_i)} \right)^{\frac{1}{m-1}} \right. \quad (7.17)$$

where

$$d^2(X_j, V_i) = (X_j - V_i)^T (X_j - V_i) \quad (7.18)$$

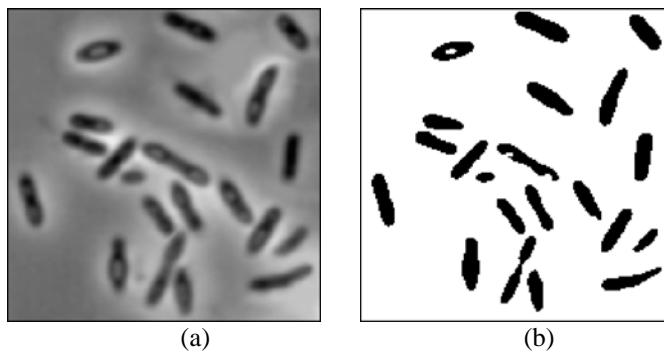
and the objective function is defined by

$$J_m = \sum_{i=1}^c \sum_{j=1}^n (\mu_{ij})^m d^2(X_j, V_i) \quad (7.19)$$

Once the goal over the objective function is met, the centroids  $V_i$  are used to classify the pixels into black or white. In this case since membership values are directly related to brightness the defuzzification function was

$$D : (0,1) \Rightarrow (0, L-1) \quad (7.20)$$

Figure 7.3 shows an example of fuzzy binarization. We can observe that although the image presents not well defined object regions the objects are correctly defined in the binarized image.



**Figure 7.3.** Fuzzy binarization: (a) Original; (b) Binary image

### 7.3.2 Fuzzy Edge Definition

Edge detection is an important operation in image processing tasks. Its purpose is to try to find objects or regions that compose the image, this task corresponding to the segmentation of an image. Edge detection has been broadly treated and documented since the early age of image processing. Although much research effort has been devoted to solving the edge detection problem it is still an open problem. The main problem on edge detection is to determine which pixels belong to the edge and which not. The complexity of the problem emerges from the fact that in general object edges are fuzzy regions. Uncertainty related to edge composition is very high in the neighborhood of the edge. The next application was developed to determine the edges in an image. The method is based on defining image level edges [12]. The level of an edge is defined based on how much attention a person must use to detect the edge. For example, edge level one corresponds to the edges that are easily detected, without too much attention, edge level two requires more attention, and edge level  $n$  requires the most attention to be detected by a person. The process is as follows.

Take an original image  $I(x, y)$ , and generate and image  $I_e$  with the gradient operator

$$\nabla : I(x, y) \rightarrow I_E \quad (7.21)$$

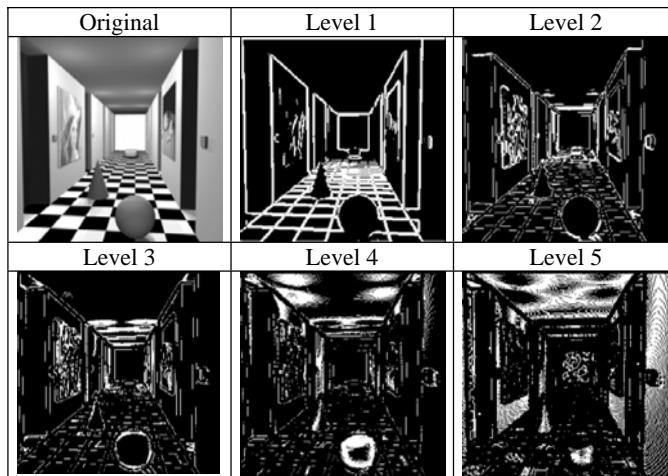
The fuzzification function is accomplished by fuzzy centroids,  $C_i$ . This function classifies the pixels of  $p_{ed} \in I_E$ , according to the following rule:

$$\begin{aligned} &\text{if } |p_{ed} - C_i| < |p_{ed} - C_j| \quad \text{for } i, j = 1, \dots, 5 \quad i \neq j \\ &\text{then } p_{ed} \in E_{li} \text{ and mark pixel } p_{ed} \text{ as level } i \end{aligned} \quad (7.22)$$

where  $E_l$  is an edge level. This classification process generates a new image  $I_{El}$  where the pixels are marked as edge level 1, 2, 3, 4 and 5. Thus, images with different edge level information can be generated from  $I_{El}$ .  $C_i$  corresponds to edge level centroids defined by the fuzzy clustering algorithm described in Section 7.3.1.

The defuzzification function in this case is implicit in the mapping to edge levels. Figure 7.4 illustrates the fuzzy decomposition of edges.

Once the edge levels are determined, they can be used to define the edges of the image [18]. For example, an automatic edge detector is implemented by computing the *complexity* of the image. Here, complexity is a measure of how complex the image is to a human observer. *Complexity* is computed by fuzzy rules using the edge level information. Table 7.1 shows some of these fuzzy rules.



**Figure 7.4.** Fuzzy edge level definition

**Table 7.1.** Fuzzy rules to determine the complexity of an image

Level 1	Level 2	Complexity	Rule No.
None	None	No complex	1
	A few	Very little 2	2
	Many	Some 2	5
	A lot	A lot 2	6
Some	None	A few1	13
	More or Less	More or Less3	16
	Many	Some 2	17
A lot	None	A lot 1	25
	A few	A lot 1	26

Using the value of complexity, the edges are defined by rules like

R1: If the image is *A little complex* then the edges are

$$\int_{i=1,2,3} I_{Ei(x,y)} \quad (7.23)$$

R2: If the image is *A lot complex* then the edges are

$$\int_{i=1,2} I_{Ei(x,y)} \quad (7.24)$$

These rules define the edge levels that are incorporated to the edges of the image. Figure 7.5 shows the result of this method.



**Figure 7.5.** Edges of the original image on Figure 7.4

### 7.3.3 Fuzzy Geometry Measurement

Many vision inspection applications require one to measure objects or attributes of the objects. We know that the measuring process by itself incorporates uncertainty. Therefore, this kind of task is suitable for fuzzy logic schemes. The following application was developed to investigate whether fuzzy processing may reduce measurement variations of geometrical characteristics, in this case area and perimeter, due to illumination.

Because of the characteristics of the problem, binary images are used in the process. The fuzzification function is the fuzzy binarization process presented in Section 7.3.1. Although fuzzy geometry is defined in [19,20], this work uses a simplification of those definitions. The fuzzy area is defined as the sum of the membership values of the object pixels.

$$\text{Fuzzy Area} = \sum_{x \in O} \mu(x) \quad (7.25)$$

where  $\mu(x)$  is the membership value of a pixel, and  $O$  is the set of pixels corresponding to the object. Values of the fuzzy area compared with the crisp area considering variations on the illumination are shown in Figure 7.6. From these experiments we can verify that the fuzzy area is less sensitive to illumination changes in this experiment. That is, the fuzzy scheme turns to be more robust to illumination variations.

In order to compute the perimeter of the image, the image is fuzzified by the fuzzy binarization algorithm. The fuzzy perimeter is defined by

$$\text{Fuzzy Perimeter} = \sum_{x \in PO} \mu(x) \quad (7.26)$$

where  $\mu(x)$  is the membership value of a pixel, and  $PO$  is the set of pixels corresponding to the perimeter of the object. Results of the calculation of the fuzzy perimeter are illustrated in Figure 7.7. It can be seen from Figure 7.7 that the fuzzy perimeter is also less sensitive to illumination changes. In this application, the defuzzification process was not necessary.

## 7.4. Results and Conclusions

In this chapter we have discussed a formal scheme to perform fuzzy image processing. This scheme is proposed as an alternative to crisp image processing algorithms, especially when subjective or very sensitive parameters or concepts related to the image need to be measured or defined.

The proposed scheme involves an image fuzzification function, fuzzy operators, and an optional defuzzification function. The applicability of the scheme is illustrated in three applications, image binarization, edge detection, and geometric measurements. On image binarization the fuzzy scheme illustrated how to resolve uncertainty on the classification of pixels to define its membership to object or background classes. The fuzzy edge detection based on the determination of image *complexity* showed how a subjective concept could be modeled by a fuzzy scheme. The last application, fuzzy geometry, is an example of how to obtain more consistent measurements under variable conditions of illumination.

Illumination	Illumination =-1	Illumination =-2
Image		
Fuzzy Area	9518.472	9893.763
Crisp Area	9536.0	9918.0
Illumination	Illumination =-3	Illumination =-4
Image		
Fuzzy Area	9837.303	9837.303
Crisp Area	9858.0	9858.0
Illumination	Illumination =-5	
Image		
Fuzzy Area	9945.954	
Crisp Area	9949.0	

Figure 7.6. Area of an object with different illumination

<b>Illumination</b>	<b>Illumination =0</b>	<b>Illumination =1</b>
Image		
Fuzzy Perimeter	297.655	303.15
Crisp Perimeter	305.0	310.0
<b>Illumination</b>	<b>Illumination =2</b>	<b>Illumination =3</b>
Image		
Fuzzy Perimeter	300.198	303.675
Crisp Perimeter	309.0	311.0
<b>Illumination</b>	<b>Illumination =4</b>	
Image		
Fuzzy Perimeter	378.341	
Crisp Perimeter	385.0	

**Figure 7.7.** Perimeter of an object with different illumination

## Acknowledgement

This work is supported in part by Dirección General de Educación Superior and COSNET grant 445.05-P.

## References

- [1] A. Ronsefeld, "Survey: Image Analysis and Computer Vision; 1993", *Computer Vision and Image Understanding*, Vo. 59, No3, May 1994, pp 367-404.

- [2] L.A. Zadeh, "Fuzzy Logic = Computing with words", *IEEE Transactions on Fuzzy Systems*, Vol. 4, No. 2, May 1996,pp 103–111.
- [3] M.M. Gupta, G.K. Knopf, and P.N. Nikiforuk, "Computer Vision with Fuzzy Edge Perception", *International Symposium on Intelligent Control*, Philadelphia, USA, 1987, pp 271–278.
- [4] R. Marks, "Fuzzy Logic Technology and Applications," *IEEE Press*, 1994.
- [5] T. Law, H. Hidenori, and H. Seki," Image Filtering, Edge Detection, and Edge Tracing Using Fuzzy Reasoning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 5, May 1996, pp 481-491.
- [6] F. Russo, " Edge Detection in Noisy Images Using Fuzzy Reasoning," *IEEE Trans. on Instrumentation and Measurement*, Vol. 47, No 5, 1998, pp 1102-1105.
- [7] Huntsberger T., Rangarajan C, "Representation of Uncertainty in Computer Vision Using Fuzzy Sets," *IEEE Trans. on Computers*, Vol. c-35, No 2, February 1986, pp 145-155.
- [8] Y.S. Choi, and R. Krishnapuram, "A Robust Approach to Image Enhancement Based on Fuzzy Logic," *IEEE Transactions on Image Processing*, Vol. 6, No. 6, pp 808 - 825, June 1997.
- [9] V. Boskovitz, and H. Guterman, "An Adaptive Neuron-Fuzzy System for Automatic Image Segmentation and Edge Detection," *IEEE Transactions on Fuzzy Systems*, Vol. 10, No 2, April 2002 , pp 247-261.
- [10] V. Lashkia, " Defect Detection in X-ray Images Using Fuzzy Reasoning", *Image and Vision Computing*, Vol. 19, 2001, pp 161-269.
- [11] M.I. Chacón, "Fuzzy Binarization and Segmentation of Text Images for OPCR," *International Conference in Signal Processing Application and Technology*, Boston Massachusetts U.S.A., 1996, pp 1091– 095.
- [12] M.I. Chacón, and L. Aguilar "A Fuzzy Approach to Edge Level Detection," *The 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, December 2001, pp 809-812.
- [13] M. I. Chacón , L. Aguilar, and A. Delgado, " Fuzzy Adaptive Edge Definition Based on the Complexity of the Image," *The 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, December 2001, pp 675–678.
- [14] M. Nachtegael, D. Weken, D. Ville, E. Kerre, "Studies in Fuzziness and Soft Computing: Fuzzy Filters for Image Processing," *Springer* 2003, pp 28-29.
- [15] G. Klir, and B. Yuan, "Fuzzy Sets and Fuzzy Logic," *Prentice Hall*, New Jersey, 1995.
- [16] T. Ross, "Fuzzy Logic with Engineering Applications", *Wiley*, 2nd ,2004, pp 96-112.
- [17] J. Bezdek,, "Pattern Recognition with Fuzzy Objective Function Algorithms," *Plenum Press*, 1981.
- [18] M.I. Chacón , and A. Corral " Image Complexity Measure: a Human Criterion Free Approach," *The 2005 North American Fuzzy Information Processing Society Annual Conference*, Ann Harvor, Michigan, USA, June 2005.
- [19] A. Rosenfeld, "The Fuzzy Geometry of Image Subsets," *Pattern Recognition Letters* 2, 1984, pp 311-317.
- [20] A. Rosenfeld, "The Perimeter of a Fuzzy Set," *Center of Automation Research, University of Maryland*, Technical Report, May 1983.

## Fuzzy Logic for Medical Engineering: An Application to Vessel Segmentation

Guillermo Ayala, Teresa León and Victoria Zapater

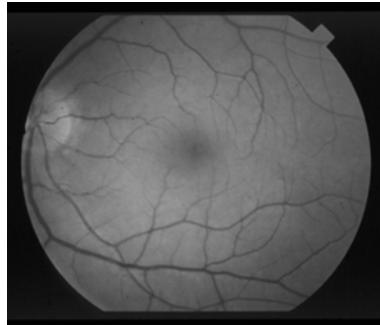
### 8.1 Introduction

The application of fuzzy sets and systems (FSS) to Medical Engineering could be fruitful both in a theoretical and a practical way. On the one hand, fuzzy logic provides the formal tools to deal with the aspects of uncertainty, vagueness and fuzziness that are inherent to most of the problems and on the other hand, the application of soft computing techniques improves the reliability and robustness of the Medical Engineering tools.

In this chapter, we are interested in the problem of retinal vessel detection in fundus images. Optic fundus assessment has been widely used by the medical community for diagnosing different pathologies such as hypertension, diabetes, arteriosclerosis, cardiovascular disease and strokes. An automatic assessment for blood vessel anomalies in the optic fundus initially requires distinguishing the vessels from the background, which is the aim of the so-called segmentation methods. Figure 8.1 displays a fundus image. This image belongs to an image database developed by Hoover and Kouznetsova [8], composed of 20 color ocular fundus images. In order to associate a gray-scale image with every color image, we have used the set of weights [0.2989, 0.5870, 0.1140], each one corresponding to an RGB band. In fact, Figure 8.1. is the gray level image obtained from the original color image. This image database can be freely downloaded from the associated Web site at: <http://www.parl.clemson.edu/stare/probing/>.

Notice that there is some ambiguity in determining the exact position of a vessel segment in the image due to the physics of the image generation process, the individual characteristics of the imaging system and the presence of noise. A number of methods have been developed for vessel segmentation and different classifications of such methods can be found in the literature. One complete review can be found in [7]. Its authors classify the vessel detection methods into five categories: matched filtering, morphological methods, artificial neural networks, vessel tracking algorithms and manual techniques. We have chosen two methods to illustrate how fuzzy logic can be helpful in improving the quality of the final segmentations. These algorithms were introduced by Zana and Klein [17] and

Heneghan et al. [7]. From now on, they will be called the Zana-Klein and Heneghan methods.



**Figure 8.1.** Fundus image from Hoover's image database (image number 81)

The Heneghan method produces a gray-scale image such that a high gray level indicates a high possibility that the pixel belongs to a vessel. Every segmentation procedure must finish with a binary image; therefore in the end each pixel is valued as one (vessel) or zero (non-vessel) by using a hysteresis thresholding whose parameters are supplied by the user. Threshold value selection will have considerable effects on the boundary position and the overall size of the extracted object. This is a problem common to many segmentation algorithms, because untrained community health workers may not feel comfortable at utilizing this technology. Our proposal is to consider the image obtained before applying the final threshold as a fuzzy set representing the vessels and then replace it by a crisp set that could be called the center or average set, which corresponds to the segmentation.

The Zana-Klein method does not use a final threshold, but the segmentation depends on a previous parameter, the length  $L$  of the linear structuring element. Again, we face the same parameter choice problem: different parameters give rise to different segmentations. The question here is conceptually different because we have different segmentations corresponding to different values of  $L$  and we propose a fuzzy method to combine them.

Figure 8.2 displays the vessels segmented (from the original image displayed in Figure 8.1) by using the methods of Heneghan and Zana-Klein, respectively. In both cases, we have chosen the best results that can be obtained with the corresponding method by performing a qualitative evaluation.

Section 8.2 gives a short description of the two segmentation methods used in this chapter. Different definitions of fuzzy set averages appear in Section 8.3. Sections 8.4 and 8.5 show how the fuzzy averages can be used to improve the segmentation procedures. In addition to the image displayed in Figure 8.1, we will use another one (see Figure 8.3) from the same image database. One important

advantage of using this database is that the authors provide two different segmentations (manually labeled vessel networks of the image set) done by two experts. This will allow us to make numerical comparisons. The chapter finishes with conclusions and some further developments in Section 8.6.

## 8.2 Two Retinal Vessel Detection Procedures

Two well-known and extensively used segmentation procedures have been implemented, namely the Heneghan and Zana-Klein methods. First of all, we would like to say that, although other procedures could be chosen, we are not primarily interested in a comparison between retinal vessel detection methods. Our interest is rather to show how the average of the different outputs obtained from them can improve the final segmentation.

Let us briefly summarize them although the original references should be consulted. Both methods use techniques taken from Mathematical Morphology. Good standard references on this topic are [12,14,16]. Mathematical Morphology is a powerful tool when the shape of the object to segment is approximately known in advance. This is the case with vessels which are connected linear structures of different lengths. The two most basic operations in mathematical morphology are erosion and dilation. Both of these operators take two pieces of data as input: an image to be eroded or dilated, and a structuring element. The two methods chosen start by emphasizing the linear structures, followed by an enhancement step using a second derivative operator. Although based on the same ideas, the implementation of these steps is different in the Zana-Klein and Heneghan methods. Finally, the Zana-Klein method applies an alternating filter to the resulting image in order to reduce gray-scale variation in the image and provides us with a segmentation based on an automatic thresholding procedure. The Heneghan method uses a similar alternating filter but the automatic thresholding is replaced by a hysteresis thresholding whose parameters are supplied by the user.

A full and detailed presentation of the concepts given in the next section can be found in [3] (see also [13]). In this chapter, we will adopt a more intuitive presentation. We are mainly concerned with the application and our aim is to explain the approach, thus opening other possible definitions.

## 8.3 Different Averages of a Fuzzy Set

First, we will introduce the main ideas of the chapter without many mathematical details and then we will state the concepts in a more formal setting. Let  $u$  be a fuzzy set in the  $d$ -dimensional Euclidean space  $\mathbf{R}^d$ , this being a non-random fuzzy set. However, we will define a random set from  $u$ . For a value  $\alpha$  in the unit interval  $[0,1]$ , let  $u_\alpha$  be the corresponding  $\alpha$ -level, *i.e.*  $u_\alpha = \{x: u(x) \geq \alpha\}$ . For a fixed  $u$  and a fixed  $\alpha$ , the  $\alpha$ -level  $u_\alpha$  will be a fixed set. But if we choose the value  $\alpha$  randomly (in accordance with some distribution in  $[0,1]$ ), then we will have a random set. This simple idea will be the cornerstone of the chapter: to associate a random set  $\Phi$

to a fixed fuzzy set  $u$  by choosing a random  $\alpha$ -level. In this way, we establish a link between the fuzzy set theory and the random closed set theory. This link could have many applications. The first application will be to associate a crisp set to the fuzzy set. The fuzzy set  $u$  is equivalent to the family of  $u_\alpha$  with  $\alpha \in [0,1]$ . If we consider a probability distribution in the unit interval  $[0,1]$  then we have a random set which is really a random  $\alpha$ -level. Different definitions of mean set exist. We can use each definition of mean set to define the corresponding mean  $\alpha$ -level. This mean  $\alpha$ -level can be a good crisp descriptor of the fuzzy set  $u$ . Different mean sets have been proposed in the literature. In our case, this is not a drawback because it opens the way for different possible definitions of mean  $\alpha$ -level.

Let us review the main definitions of mean sets proposed. Some basic definitions must be given. Let us begin with the definition of a random closed set. This concept was formally defined in [10]. Let  $\mathbf{F}$  be the family of the closed sets in  $\mathbb{R}^d$ . Let us consider, for a compact set  $K$ , the set  $\mathbf{F}_K$  that is composed by the closed sets touching  $K$ . Let  $\sigma_f$  be the  $\sigma$ -algebra generated by the sets  $\mathbf{F}_K$  where  $K$  is an arbitrary compact set. If  $P$  is a probability measure in the measurable space  $(\mathbf{F}, \sigma_f)$ , then a random closed set is the probability space  $(\mathbf{F}, \sigma_f, P)$ . In summary, a random closed set is the (natural) probabilistic model when the realizations are closed subsets of  $\mathbb{R}^d$ .

Given a random  $\Phi$ , there is no unique definition of mean set  $E\Phi$  associated to  $\Phi$ . Different definitions of the mean set of a random compact set can be found in the literature, three of which are particularly important: the Aumann mean, the Vorob'ev mean and the Baddeley-Molchanov mean. In this section a brief review of their definitions is given. However, a more detailed explanation appears in [15] for the Aumann and Vorob'ev means and [5] for the Baddeley-Molchanov mean.

Let  $\Phi$  be a random compact set on  $\mathbb{R}^d$ , *i.e.* a random closed set taking values in the space  $\mathbf{K}$  of all compact subsets of  $\mathbb{R}^d$ . The Aumann definition of a mean set is based on the concept of support function. Given a fixed set  $X$ , its support function is defined as  $h(X, u) = \sup \{ u^*x : x \in X \}$  where  $u^*x$  denotes the inner product of  $u$  with  $x$  and  $u$  runs over the unit circumference centred at the origin. If  $\Phi$  is a random compact set, then  $h(\Phi, \cdot)$  is a random function. The Aumann mean of  $\Phi$ ,  $E_a\Phi$ , is defined as the convex set with support function given by the mean or expected value of the random variable  $h(\Phi, u)$ , *i.e.*  $h(E_a\Phi, u) = E h(\Phi, u)$ .

This definition is equivalent to the one originally put forward in [2] and was proposed in [1]. Let us now look at the sample estimator of this mean set.

From now on,  $\Phi_i$  with  $i=1, \dots, n$  will denote a random sample, *i.e.* independent and identically distributed (as  $\Phi$ ) random sets. In accordance with [15],  $E_a\Phi$  can be estimated by means of

$$\hat{E}_a \Phi = (\text{co}(\Phi_1) \oplus \dots \oplus \text{co}(\Phi_n)) / n$$

where  $\text{co}(\Phi_i)$  is the convex hull of the set  $\Phi_i$ .

Vorob'ev (the definition is shown in [15], the original reference of Vorob'ev was written in Russian) proposed a different definition of mean set based on a thresholding of the coverage function of  $\Phi$ . The coverage function is defined, at point  $x$ , as the probability of this point belonging to the random set  $\Phi$ ,  $P(x \in \Phi)$ . Let  $S_p = \{x \in \Phi : P(x \in \Phi) \geq p\}$ . The Vorob'ev mean set  $E_v\Phi$  is defined as  $S_{p'}$  where  $p'$  is

chosen in such a way that the area of  $S_p$  is equal to the mean area of  $\Phi$ . If different  $p$  values verify the equality then the minimum is used. Given a sample of  $\Phi$ ,  $E_v\Phi$  is naturally estimated starting from the sample coverage function defined as

$$\hat{P}(x \in \Phi) = \frac{1}{n} \sum_{i=1}^n 1_{\Phi_i}(x), \text{ and}$$

$$\hat{S}_{\hat{p}} = \{x \in \Phi : \hat{P}(x \in \Phi) \geq \hat{p}\},$$

$$\text{where } v(\hat{S}_{\hat{p}}) = \frac{1}{n} \sum_{i=1}^n v(\Phi_i)$$

Finally, Baddeley and Molchanov proposed a new concept of mean set based on their distance function representation. A good and detailed presentation with many examples can be found in the original reference [5]. Let  $\mathbf{F}'$  be the space of non-empty closed sets with hit-miss topology (see [10]) and let  $d: \mathbf{R}^n \times \mathbf{F}' \rightarrow \mathbf{R}$  be a generalized distance function, *i.e.* a lower semi-continuous function with respect to the first argument and measurable with respect to the second argument in such a way that the two following requirements are fulfilled: (i) if  $F_1 \subseteq F_2$  then  $d(x, F_1) \leq d(x, F_2)$  for all  $x$  in  $\mathbf{R}^n$ , and (ii)  $F = \{x : d(x, F) \leq 0\}$ . Let  $m$  be a metric (or pseudometric) on the family of distance functions and  $m_W$  the restriction of  $m$  to  $W$ , a certain compact set (window). Suppose that  $d(x, \Phi)$  is integrable for all  $x$  and we define the mean distance function  $d^*(x) = E_d(x, \Phi)$ . Let  $\Phi(\varepsilon) = \{x \in W : d^*(x) \leq \varepsilon\}$  with a positive  $\varepsilon$  and let  $\varepsilon^* = \operatorname{argmin}_\varepsilon m_W(\Phi(\varepsilon), d^*)$ . The Baddeley-Molchanov mean of  $\Phi$ ,  $E_{bm}\Phi$ , is the set  $\Phi(\varepsilon^*)$ .

The estimation of  $E_{bm}\Phi$  results from the empirical distance average defined as

$$\hat{d}^*(x) = \frac{1}{n} \sum_{i=1}^n d(x, \Phi_i).$$

It is worth noting at this point that the Aumann mean is only suitable for averaging convex sets, *i.e.* for random compact and convex sets. Concerning the Vorob'ev mean, although it is applicable to non-convex sets, it is not suitable for random sets with zero area and it is extremely sensitive to small displacements and errors. On the other hand, it must be noted that whereas the Aumann and Vorob'ev mean sets are unique, the Baddeley-Molchanov mean depends on the distance chosen.

There is a connection between the Aumann mean of a random compact set as has been previously defined and the average level of a fuzzy set defined in [11]. Now,  $u$  is a (non-random) fuzzy set and  $\Phi$  is a random compact set defined as follows: a uniform value  $\alpha$  is generated in the unit interval  $[0, 1]$  and given  $\alpha$ ,  $\Phi = u_\alpha$ . It will be assumed that the set  $u_\alpha$  is closed for any  $\alpha$ . In fact, we assume that  $u$  is an upper semi-continuous function. If  $\Phi$  is defined in this way, then the Aumann mean of this random set (see [5]) coincides with the average level proposed in Ralescu (2000) [11]. It is not difficult to generalize this definition.

**Definition 1** The Ralescu-Aumann fuzzy set average

Let  $u$  be a fuzzy set and let  $\Phi$  be a random compact defined as  $\Phi = u_\alpha$ , where  $\alpha$  is a random value in  $[0,1]$  with a probability distribution  $P_i$ ; then the Ralescu-Aumann fuzzy set average  $A_{ra}(u)$  is the convex crisp set with its support function

$$\text{given by } h(A_{ra}(u), v) = E_\Phi h(\Phi, v) = \int_0^1 h(u_\alpha, v) P_i(d\alpha)$$

As we have seen, a kind of mean taken from random compact set theory has been used to define an average associated with a given fuzzy set. Therefore, it seems natural and a promising line of research to use the same procedure with some other definitions of a mean of a random set.

Note that the Aumann mean and its translation into fuzzy sets, the Ralescu-Aumann fuzzy set average, have a serious drawback. Although the different  $\alpha$ -level sets  $u_\alpha$  are not convex sets, the corresponding mean set is convex. Note that, for instance, in vessel segmentation the objects to be detected are clearly non-convex. Two particularly important definitions of mean sets which avoid this question are the Vorob'ev mean and the Baddeley-Molchanov mean.

**Definition 2** The Vorob'ev fuzzy set average

Let  $u$  be a fuzzy set and let  $\Phi$  be a random compact defined as:  $\Phi = u_\alpha$  where  $\alpha$  is a random value in  $[0,1]$  with a probability distribution  $P_i$ . The Vorob'ev fuzzy set average  $u$ ,  $A_v u$ , is defined as  $A_v u = u_{\alpha^*}$ , where  $v(u_{\alpha^*}) = \int_{[0,1]} v(u_\alpha) P_i(d\alpha)$

If different  $\alpha^*$ -levels verify the former equation, the minimum is taken.

Baddeley and Molchanov also proposed a mean set based on its distance function representation. A good and detailed presentation can be found in the original reference [5].

Suppose that  $d(x, \Phi)$  is integrable for all  $x$  and we define the mean distance function  $d^*(x) = E_d(x, \Phi)$ . Let  $\Phi(\varepsilon) = \{x \in W: d^*(x) \leq \varepsilon\}$  with  $\varepsilon \in \mathbb{R}$  and let  $\varepsilon^* = \text{argmin}_\varepsilon m_W(\Phi(\varepsilon), d^*)$ . The Baddeley-Molchanov mean of  $\Phi$ ,  $E_{bm}\Phi$  is the set  $\Phi(\varepsilon^*)$ . Again, the  $\alpha$ -level sets will replace the realizations in order to define the fuzzy set average.

**Definition 3** The Baddeley-Molchanov fuzzy set average

Let  $u$  be a fuzzy set and let  $\Phi$  be a random compact defined as:  $\Phi = u_\alpha$  where  $\alpha$  is a random value in  $[0,1]$  with probability distribution  $P_i$ . Let  $d$  be a generalized

distance function verifying all previous assumptions. If  $\hat{d}_u(x) = \int_0^1 d(x, u_\alpha) P_i(d\alpha)$

and  $N_\varepsilon = \{x: \hat{d}_u(x) \leq \varepsilon\}$ .

The Baddeley-Molchanov fuzzy set average,  $A_{bm}u$ , is defined as  $N_{\varepsilon^*}$  where

$$\varepsilon^* = \text{argmin} D(1_{N_\varepsilon}, \hat{d}_u)$$

where  $1_A$  is the indicator function of  $A$  and  $D$  is a metric between functions.

The optimal  $\varepsilon'$  in this definition is not easy to obtain. For such reasons the modified definition of the Baddeley-Molchanov mean [9] is usually preferred and we have also adapted it for fuzzy sets.

**Definition 4** The modified Baddeley-Molchanov fuzzy set average

Let  $u$  be a fuzzy set and let  $\Phi$  be a random compact defined as  $\Phi = u_\alpha$  where  $\alpha$  is a random value in  $[0,1]$  with probability distribution  $P_l$ . Let  $d$  be a generalized

distance function verifying all previous assumptions. If  $\hat{d}_u(x) = \int_0^1 d(x, u_\alpha) P_l(d\alpha)$

and  $N_\varepsilon = \{x: \hat{d}_u(x) \leq \varepsilon\}$ , the modified Baddeley-Molchanov fuzzy set average  $u$ ,  $A_{mbm} u$ , is defined as  $N_\varepsilon$  where

$$v(N_\varepsilon) = \int_0^1 v(u_\alpha) P_l(d\alpha)$$

In order to study how the Vorob'ev fuzzy set average works we have considered an interesting class of fuzzy sets, the L-R fuzzy numbers [3]. Actually, we have compared the well-known mean set of Dubois and Prade [6] with the Vorob'ev fuzzy set average checking the consistency of the new concept. Both definitions provide us with the same or very similar sets, depending on the cases, as we expected.

Our idea is to combine a segmentation procedure with a fuzzy set average. Preliminary studies have shown us that the best combinations are provided by the Heneghan method with the Vorob'ev fuzzy set average and by the Zana-Klein method with the modified Baddeley-Molchanov fuzzy set average. The following sections are devoted to describing the results obtained using such associations.

The comparison between different segmentation procedures applied to vessel detection in fundus images is not a simple task with a unique numerical solution. As we have previously mentioned, each image in the Hoover database has been segmented by two experts. It will permit us to compare the unsupervised segmentation procedures with these supervised segmentations. Pixels that belong to an expert segmentation but not to our segmented image will be called false negatives and the corresponding count will be denoted as  $eII$ . If a given pixel belongs to our segmented image, but the expert does not consider it as a vessel, then we have a false negative and the corresponding count is denoted as  $eI$ . A given (automatic) segmentation will be compared with an expert by using the ratio

$$E = (eI + eII) / A,$$

where  $A$  is the area of the expert segmentation.

## 8.4 The Heneghan Method with the Vorob'ev Fuzzy Set Average

Let us recall that the Heneghan method finishes with a threshold of the gray level image. It is important to take into account that the Vorob'ev average is a particular  $\alpha$ -level. Therefore, it could be considered as an automatic threshold. In particular, for small  $\alpha$  values the corresponding  $\alpha$ -level sets will contain practically the whole retina. In contrast, if we consider large  $\alpha$  values then the  $\alpha$ -level sets will correspond to the median axis of the large vessels emerging from the optic disk.

In the previous section, we presented the Vorob'ev fuzzy set average in a general framework with a probability distribution on the unit interval  $[0,1]$ . Now we will specify a particular distribution in order to apply the average procedure to vessel detection. The beta distributions are the natural choice. They are defined on  $[0,1]$  and, depending on their parameters, the shapes of the corresponding density functions are very different. The probability density function of a beta distribution with shape parameters  $a$  and  $b$  is given by

$$f(\alpha) = 1/B(a, b) \alpha^{a-1} (1-\alpha)^{b-1}, \quad 0 \leq \alpha \leq 1,$$

where the beta function is given by

$$B(a, b) = \int_0^\infty x^{a-1} (1-x)^{b-1} dx.$$

In practice, a discrete set of  $\alpha$  values  $\alpha_1, \dots, \alpha_k$  has to be used. In particular,  $0.05, \dots, 0.95$  has been used with  $k = 19$  and the resulting distribution is a discrete beta with a probability function

$$\frac{f(\alpha_i)}{\sum_{j=1}^n f(\alpha_j)}.$$

We have obtained the beta density which performs better for the Heneghan method. Its parameters have been calculated by solving an optimization problem: the sum of the values of the error  $E$  for the different experts and images in the study has been computed for all possible pairs of  $(a, b)$  with  $a, b \in \{1, 2, \dots, 30\}$  and the minimum and the corresponding parameters are retained. When different pairs  $(a, b)$  are possible then we choose those with the highest variance, looking for a smoother segmentation. Table 8.1 shows the errors obtained by using the Heneghan method combined with the Vorob'ev fuzzy set average. The rows named Modified-Vor and Original display the  $E$  values obtained with the Heneghan plus Vorob'ev method and with the Heneghan method respectively. The values of the parameters of the beta distribution chosen are  $(2, 4)$ . We would like to make the following remarks. First, this procedure to select the parameter values cannot be used in a real problem because we usually have no expert segmentation. Second, there is a great variability in the errors observed due to the inter-expert variability. Actually, one of the segmentations is more conservative and almost contains the other one. It is possible to find a pair  $(a, b)$  of parameters for each combination image-expert such that the modified procedure has a lower error than the

corresponding original. However, our purpose is not this, but rather to provide a fully robust automatic methodology. Other images from the Hoover database and other databases (angiographic images and color images) have been evaluated with the same pair of parameters and we have obtained very good results. In the worst case, the results are similar to the originals.

Observed errors using the Heneghan method and the Vorob'ev fuzzy set average. The shape parameters of the beta distribution have been set to (2, 4).

**Table 8.1.** Errors obtained by using the Heneghan method combined with the Vorob'ev fuzzy set average

	First expert		Second expert	
	im0081	im0082	im0081	im0082
<b>Modified-Vor</b>	0.4599	0.4325	0.5776	0.5230
<b>Original</b>	1.2483	0.9459	0.6199	0.6328

The evaluation of the performance of a segmentation procedure must not be based on simply using some numerical error measures [4]. In fact, the ocular inspection of the segmentations suggests that using the averaging procedures greatly improves the methods. This can be appreciated by comparing the original image in Figure 8.1 with the two segmentations displayed in Figure 8.2a and c. The large vessels are detected by both methods but the segmentation obtained by the Heneghan method declares too many pixels as a vessel and the boundaries are not smooth. Essentially, the segmentation obtained by the original method is improved without any human intervention.

## 8.5 The Zana-Klein Method and the Fuzzy Set Averages

This section studies how to average the final outputs of the Zana-Klein method by using the Vorob'ev and the Baddeley-Molchanov averages. Although both of them take into account all the  $\alpha$ -level sets, their main difference is that the Baddeley-Molchanov fuzzy set average is not generally a particular  $\alpha$ -level set.

Again, the probability distribution specified over the indices  $\alpha$  belongs to the discrete beta family. The shape parameters have been determined using the procedure described in Section 8.4 and we have obtained the same pair of parameters ( $(a, b) = (4, 3)$ ) for both averages. We have checked that the choice of the parameter values depends on the segmentation procedure. The errors observed appear in Table 8.2. There is a clear improvement with respect to the first expert when the original and the modified Zana-Klein methods are compared. The results are very close when the second expert is considered. Again, if a different pair of parameters is used for each combination image-expert then the differences would be larger (in favor of the modified version). Figure 8.2b displays the best segmentation obtained by the Zana-Klein method with the expert supervision. Figure 8.2d shows the automatic unsupervised segmentation when the Baddeley-Molchanov fuzzy average replaces the expert.

The main vessels are preserved. Some pieces of small vessels are then removed by the modified method. In our opinion they correspond with unconnected small vessels. The foveal avascular zone or FAZ (dark rounded area in the center of the retina) contains some false vessels detected with the original method. No vessel should be detectable in this avascular zone. The modified method has no false positives in this area.

The combination between the Baddeley-Molchanov average and the Heneghan method has provided us with bad results from quantitative and qualitative points of view. We are not really sure of the reasons. Note that there is a fundamental difference between the membership functions defined for the Heneghan and the Zana-Klein methods. In the Zana-Klein membership function different segmentations are combined as the average at each pixel, *i.e.* we have segmentations and combine them. This could possibly be a partial explanation. Another possibility is that the distance chosen for us is not suitable for these kinds of images. However, the combination between the Zana-Klein method and the Baddeley-Molchanov average provides the best global results.

Besides this example, Figure 8.3 displays the second image analyzed. Tables 8.1 and 8.2 contain the errors observed and the corresponding segmentations appear in Figure 8.4. Similar comments to the first example can be made.

Observed errors using the Zana-Klein method and the Vorob'ev and Baddeley-Molchanov fuzzy set averages. The shape parameters of the beta distribution have been set to (4, 3).

**Table 8.2.** Errors obtained by using the Zana-Klein method and the modified methods by incorporating the Vorob'ev and the Baddeley-Molchanov means

	First expert		Second expert	
	im0081	im0082	im0081	im0082
<b>Modified-Vor</b>	0.4886	0.4522	0.4864	0.4645
<b>Modified B-M</b>	0.4666	0.4418	0.4782	0.4576
<b>Original</b>	0.6034	0.5360	0.4886	0.4781

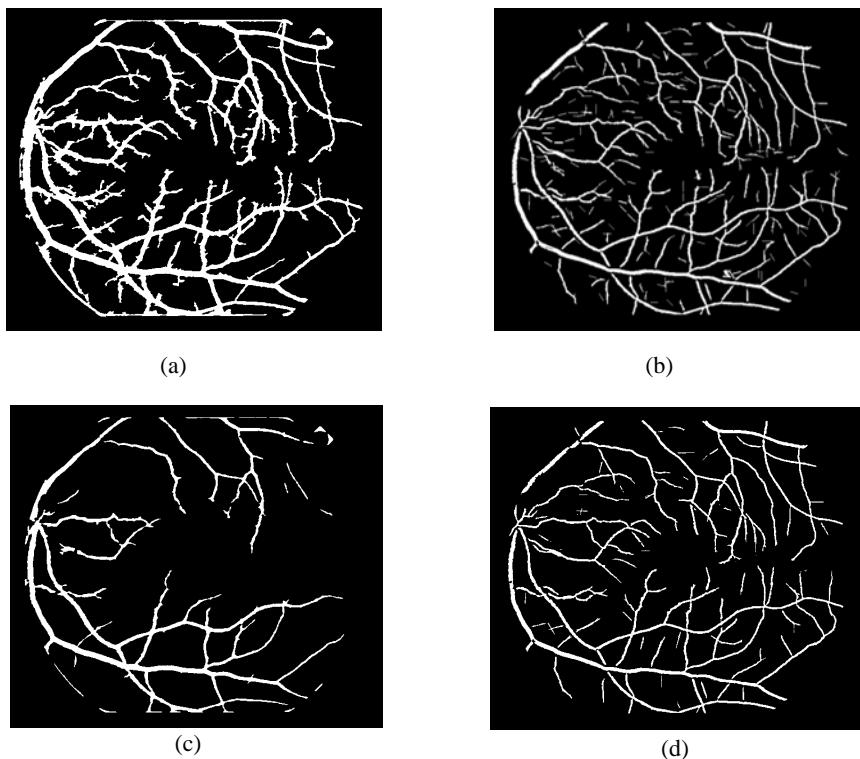
## 8.6 Some Conclusions and Further Developments

Image segmentation is a major problem in Image Processing, particularly in Medical Image Analysis. A great number of segmentation procedures produce intermediate gray-scale images that can be understood as fuzzy sets. Additionally, some segmentation procedures tend to leave free tuning parameters (very influential in the final binary image) for the user. These different binary images can be easily aggregated (into a fuzzy set) by making use of fuzzy set theory. In any case, a single binary image is required so our interest is to associate a crisp set to a given fuzzy set in an intelligent and unsupervised manner. Two new fuzzy set averages have been proposed. Differently to other proposals in the fuzzy set literature, these averages are defined in an arbitrary  $d$ -dimensional Euclidean

space. This is a basic advantage of our approach with respect to the well-known Dubois and Prade mean in order to apply them in fuzzy image processing.

We have used these new averages for vessel detection in fundus images. Two segmentation procedures have been taken from medical imaging literature. Different natural membership functions have been associated with the segmentation procedures. The fuzzy set averages provide the segmentation in an automatic manner without any human intervention. Very promising results have been achieved. The original methods become more robust and accurate.

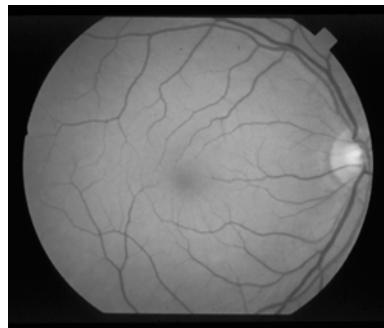
On the other hand, we would like to point out that the averages can be used at any step of any image processing procedure in which the transition from a gray-scale image to a binary image is required. This could give rise to new image processing algorithms. In particular, other vessel detection methods proposed in the literature should be considered, combined and enriched with the use of these averages.



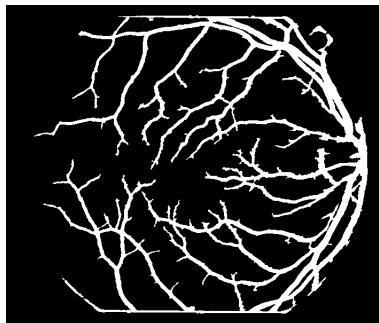
**Figure 8.2.** Segmentations of image in Figure 8.1 by using the method of: Heneghan (a), Zana-Klein (b), the combination of Heneghan and Vorob'ev (c) and the combination of Zana-Klein and Baddeley-Molchanov (d)

Many new problems arise, such as the combination of the fuzzy sets obtained from different segmentation procedures and the extension to random fuzzy sets or

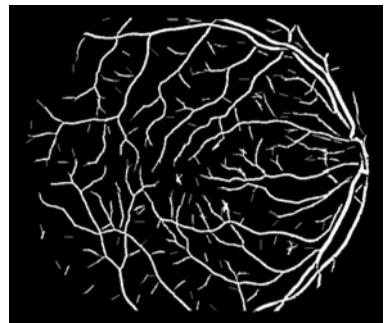
fuzzy random variables. We think that all these problems can be dealt with by using extensions of the proposed averages.



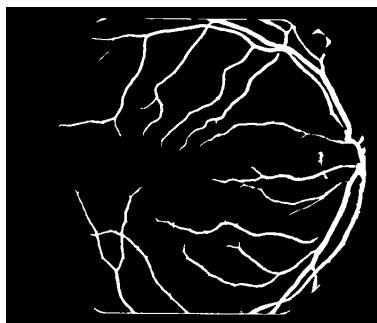
**Figure 8.3.** Original image (image 82)



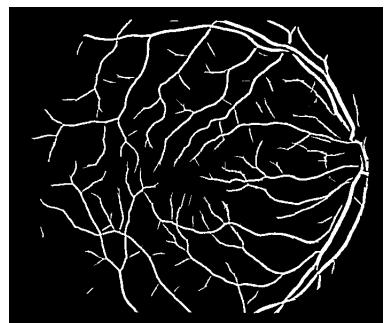
(a)



(b)



(c)



(d)

**Figure 8.4.** Segmentations of image in Figure 8.3 by using the method of: Heneghan (a), Zana-Klein (b), the combination of Heneghan and Vorob'ev (c) and the combination of Zana-Klein and Baddeley-Molchanov (d)

## Acknowledgement

This paper has been supported by grants TIC2002-03494 (Ministerio de Educación y Ciencia, Spain), RGY 40/2003 and Grupos04-08 (Generalitat Valenciana).

## References

- [1] Z. Arstein and R. A. Vitale, "A strong law of large numbers for random compact sets", *Ann. Probab.*, Vol. 3, 1975, pp 879-882.
- [2] R.J. Aumann, "Integrals of set-valued functions", *J. Math. Anal. Appl.*, Vol.12, 1965. pp 1-12.
- [3] G. Ayala, T. León and V. Zapater, "Different Averages of a Fuzzy Set With an Application to Vessel Segmentation", *IEEE Transactions on Fuzzy Systems*, Vol. 13, No. 3, 2005, pp 384-393.
- [4] A. J. Baddeley, "Errors in binary image and an  $l^p$  version of the Haussdorf metric", *Nieuw Archief Voor Wiskunde*, Vol. 10, 1992, pp 157-183.
- [5] A. J. Baddeley and I. S. Molchanov, "Averaging of random sets based on their distance functions", *J. Math. Imag. Vision*, Vol. 8, 1998, pp 79-92.
- [6] D. Dubois and H. Prade, "The mean value of a fuzzy number", *Fuzzy Sets and Systems*, Vol. 24, 1987, pp 279-300.
- [7] C. Heneghan, J. Flynn, M. O'Keefe and M. Cahill, "Characterization of changes in blood vessel width and tortuosity in retinopathy of prematurity using image analysis", *Med. Image Anal.*, Vol. 6, 2002, pp 407-429.
- [8] A. Hoover and V. Kouznetsova, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response", *IEEE Transactions on Medical Imaging*, Vol. 19, No. 3, 2000, pp 203-210.
- [9] T. Lewis, R. Owens and A.J. Baddeley, "Averaging feature maps", *Pattern Recognition*, Vol. 32, No. 3, 1999, pp 1615-1630.
- [10] G. Matheron, *Random sets and Integral Geometry*. London: Wiley, 1975.
- [11] D. Ralescu, "Average level of a fuzzy set", in *Proc. 8th Int. Conf. IPMU Information Processing and Management of Uncertainty in Knowledge-Based*, Vol. I, pp 190-194, 2000.
- [12] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic Press, 1982.
- [13] A. Simó, E. de Vés and G. Ayala, "Resuming shapes with applications", *Journal of Mathematical Imaging and Vision*, Vol. 20, 2004, pp 209-222.
- [14] P. Soille, *Morphological Image Analysis. Principles and Applications*, 2nd ed. New York: Springer-Verlag, 2003.
- [15] D. Stoyan and H. Stoyan, *Fractals, Random Shapes and Point Fields. Methods of Geometrical Statistics*. New York: Wiley 1994.
- [16] L. Vincent, *Stochastic Geometry Likelihood and Computation*. Boca Raton, FL: Chapman Hall/CRC, Vol. 80, Monographs on Statistics and Applied Probability, 1999, ch. 6, pp 199-283.
- [17] F. Zana and J.C. Klein, "A Multi-Modal Registration Algorithm of Eye Fundus Images Using Vessels Detection and Hough Transform", *IEEE Transactions on Medical Imaging*, Vol. 18, No. 5, 1999, pp 419-428.

## **Fuzzy Logic for Transportation Guidance: Developing Fuzzy Controllers for Maintaining an Inter-Vehicle Safety Headway**

Jose E. Naranjo, Carlos González, Ricardo García and Teresa de Pedro

### **9.1 Introduction**

Intelligent Transportation Systems (ITS) are systems utilizing multidisciplinary technologies to improve all kinds of transportation. One of the most important related technologies is computer science. Computers are used in the automotive industry to perform certain guidance actions, attempting to minimize injuries and to prevent collisions [1]. One application of ITS is to provide assistance for controlling some vehicle components, like speed. This is known as the cruise control (CC) system. It consists of maintaining the vehicle speed at a user (driver) pre-set speed. Such systems are already mass installed in top-of-the-line-end vehicles Adaptive cruise control (ACC) is a cruise control extension [2].

ACC is similar to conventional cruise control in that it maintains the vehicle at a pre-set speed. However, unlike conventional cruise control, this system can automatically adjust speed in order to keep a proper headway distance between vehicles in the same lane [3]. This is achieved through a radar or laser headway sensor, digital signal processor and a speed controller. If the preceding vehicle slows down, or if another object is detected, the system sends a signal to the engine or braking system to decelerate. Then, when the road is clear, the system will re-accelerate the vehicle back to the set speed [4]. Previous research has shown that ACC can improve traffic conditions significantly [5]. ACC systems have been on the market since Mitsubishi released the “Preview Distance Control” for its Diamante model car in 1995. Toyota, Nissan, Jaguar, Mercedes (Distronic), Lexus, BMW [6] and some car component industries have introduced ACC systems, although only as an optional device for luxury vehicles. One limitation of these commercial systems is that they control the speed of the car at speeds above 30-40 km/h only and do not work at lower speeds in heavy traffic, where if the preceding car stops, the equipped car must do so too at a safe headway. Stop & Go systems automate the throttle control in this kind of situation. Along these lines, the Volkswagen group recently published [7] that “The next-generation ADC+F2S (F2S = follow to stop) system will be able to bring the vehicle to a complete stop automatically behind the car ahead if necessary”.

There are a lot of techniques for performing ACC. Conventional methods based on analytical control produce good results. Nevertheless, they have high design and computational costs since the application object, a car, is a non-linear element and cannot be completely represented mathematically [8-11]. Other ways to achieve human-like speed control include the application of artificial intelligence techniques [12, 13]. One of these techniques, fuzzy control, allows approximate human reasoning and an intuitive control structure [14].

Fuzzy logic is a powerful, albeit somewhat controversial, technique. It permits control without extensive knowledge of the equations of the process and it very effectively represents human reasoning methods [15]. This technique was created by L. A. Zadeh, at Berkeley [16], and the first real application to a control system was proposed by E. H. Mamdani [17]. Nowdays, fuzzy logic is used to manage a wide range of systems, like cars [18], aircraft [19] and railways [20].

Stop and go is one of the most tedious and tiring operations human drivers have to perform, and it is also one of the most common because there are traffic jams in most cities all over the world. Accidents involving rear-end collisions are common in this kind of situations. Stop&Go systems are being developed in order to automate this maneuver. The combination of ACC and Stop&Go increases driving comfort, smooths traffic speed and allows queues to discharge faster from bottlenecks [3, 21, 22].

This chapter addresses the integration of mechatronics and fuzzy control techniques in order develop unmanned vehicle driving systems, and presents the experiments run to test this integration. The present application is for a car computer throttle control powered by a fuzzy logic controller, with the capability of performing adaptive cruise control in unmanned/manual driving. The speed tracking experiments have been conducted on a private test circuit, and the testbed cars are two automated mass-produced vehicles.

## 9.2 The Autopia Program

The Autopia Program is a set of research projects conducted by the Instituto de Automática Industrial (IAI). The IAI is a part of the Spanish Council for Scientific Research (CSIC). Building on the extensive experience of the IAI in the development of autonomous robots and fuzzy control, AUTOPIA was set up similarly to other countries' programs [23-27]. The goal of AUTOPIA is to transfer autonomous mobile robot control technologies to computer-aided vehicle driving. The aim of this program is to develop a testbed infrastructure for experimentation in control systems, strategies and sensors applied to vehicle driving, open to groups interested in our research field.

As part of AUTOPIA, three testbed cars (Figure 9.1) have been automated in order to perform lateral and longitudinal control from a computer in which a fuzzy logic-based control system is embedded. The main sensor is a high precision GPS, and the steering and the accelerator pedals of the cars have been automated. Braking has also been automated, although we will not deal with this question here.



**Figure 9.1.** Autopia autonomous vehicles

### 9.2.1 Application of a Fuzzy Coprocessor to Automatic Vehicle Driving

Fuzzy logic has been selected because it is a well tested method for dealing with very complex systems, provides good results, can incorporate human procedural knowledge into control algorithms and does not require knowledge of the mathematical model of the processes involved. The relationship between input and output variables is expressed in sentences that closely mimic human thinking.

The first step towards emulating human driving is to represent human approximate reasoning [28-31]. In [30], approximate reasoning or fuzzy reasoning is defined as the process or processes by which a possibly imprecise conclusion is deduced from a collection of imprecise premises. Previous knowledge of driver behavior can be used to extract an approximation of the fuzzy control inputs, outputs, membership functions and fuzzy rules [31]. In a second step, the fuzzy variable membership functions can be fine tuned according to experimental observation. The third step is to automate the mass-produced cars, instrumented to allow an automatic fuzzy control of the steering and the accelerator-brake set.

The fuzzy controllers are supported by ORBEX (Experimental Fuzzy Coprocessor) software [32], in which fuzzy rules can be written as sentences in near-natural language.

Two independent controllers have been designed to perform automatic driving: lateral control and longitudinal control or, in other words, steering control and speed control, respectively [33].

We will describe the longitudinal – speed – control, and the results achieved when it was installed in an automated car.

### 9.2.2 The Longitudinal Control System

Conceptually, there are three control levels in human driving for managing the accelerator of a traditional mass-produced car: a mechanical layer (the accelerator pedal and its associated mechanisms), an actuation layer (the human foot that steps on the pedal) and a knowledge reasoning layer (the human brain).

The testbed cars are Citroën Berlingo electrical vans, equipped with an automatic gearbox in order to simplify gear changing. In these cars, the throttle signal is fully electronic, and a voltage signal is sent directly from the pedal to the internal vehicle motor calculator, which makes the motor generate a proportional torque.

Similarly, when we automate this car component, we also divide the control architecture into three layers:

- A mechanical layer, which is identical to the human control level, because the car components cannot be changed. The difference is that the input is not a mechanical pressure on the pedal, but an electrical signal simulating the pedal voltage output.
- An electronic layer that, like the driver's foot, generates a signal proportional to a target speed. Two components do the job of the human foot. The first component is an industrial PC running the control software. The second component is an analog output card that mimics the electronic throttle signal. We also acquire the car speed signal with a sampling rate of 10 Hz and an absolute error of  $\pm 0.25 \text{ m/s}$ .
- Finally, a control layer made up of a fuzzy control system.

Having defined the system actuators and the electronic environment, we proceeded with the definition of the intelligent control system used for testbed car longitudinal control. In this chapter, we focus on the description of the fuzzy controller, which carries out the tasks of maintaining the speed (cruise control) and adapting to the lane speed (adaptive cruise control).

Briefly, our fuzzy controller uses the t-norm minimum to implement the AND operator. The t-conorm maximum is used to implement the OR operator. The syntax of the fuzzy rules is defined as

IF X [OR/AND Z] THEN Y

where X and Z are fuzzy input variables and Y is the fuzzy output variable. The center of mass method implements the defuzzification operation [34], as shown in Equation (9.1).

$$x_{\text{out}} = \frac{\sum_i w_i x_i}{\sum_i w_i} \quad (9.1)$$

where  $w_i$  represents the value of the weight of each rule  $i$ , and  $x_i$  is the crisp value of the condition of each rule  $i$ , weight being understood as the extent to which the crisp current values of the inputs satisfy the rule set conditions.

A fuzzy controller imitating a classical PD was defined to achieve speed control. Initially, we developed a CC fuzzy system. After that, and based on the same controller, we added adaptive capabilities to this controller to achieve an ACC system.

#### 9.2.2.1 Design of the Fuzzy Cruise Control System

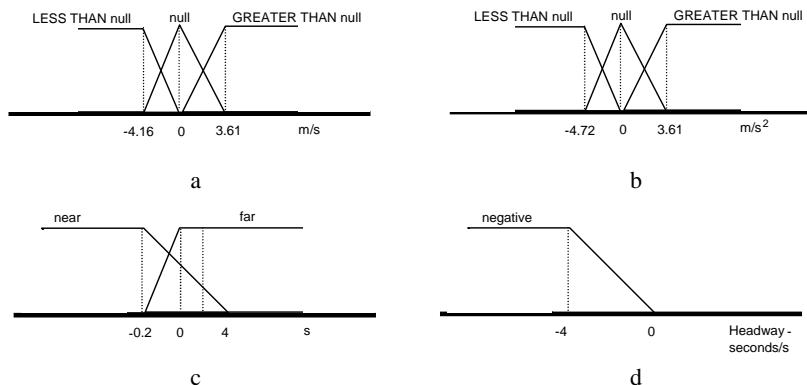
The equipped testbed vehicle provides the necessary data to gather the input information for the control system: the instantaneous speed and the time interval between two speed measures. This input consists of two variables:

Speed Error (SE) is the difference between the current speed (CS) and the user-preset speed (PS). We can express it mathematically as follows in Equation (9.2).

$$SE = CS - PS \quad (9.2)$$

Acceleration (A) is the evolution of the speed during a time interval ( $\Delta t$ ), described in Equation (9.3).

$$A_t = \frac{CS_t - CS_{t-1}}{\Delta t} \quad (9.3)$$



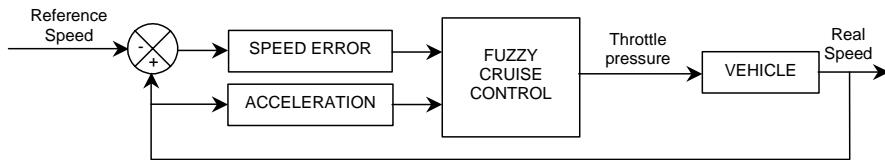
**Figure 9.2.** Fuzzy input variable membership functions

Because this calculation is somewhat noisy, we added a Fourier filter [35] to stabilize it.

The fuzzy controller will thus be formed by two fuzzy variables, named *speed\_error* and *acceleration*, whose associated linguistic labels and membership functions are illustrated in Figure 9.2a and b, where it is shown that the membership functions are not symmetrical. The dynamics of the vehicle and the mechanical actuation when the car brakes or accelerates are the reasons for this asymmetry.

The output of the fuzzy controller will be the accelerator-pedal-pressure-like signal (electrical voltage) that will be sent to the car's internal on-board computer

through the electronic layer. The fuzzy PD controller can be outlined as shown in the diagram in Figure 9.3.



**Figure 9.3.** Fuzzy cruise control architecture

The controller output is the pressure applied to the accelerator pedal, performed by an electrical signal. The fuzzy controller defines an output variable named *throttle* as two singletons: the “Up” label means the controller must release the accelerator pedal and the “Down” label means the controller must step on this pedal. This output is used as an increment and added to the accumulated value of the accelerator in each control loop iteration.

Finally, we define the fuzzy rules of the CC control system. Only four rules are necessary (Table 9.1).

**Table 9.1** Fuzzy control rules of the CC control system

		Input fuzzy variables	Linguistic modifiers	Linguistic labels		Output fuzzy variable	Linguistic Labels
R1	IF	<i>Speed_Error</i>	GREATER THAN	Null	THEN	<i>Throttle</i>	Up
R2	IF	<i>Speed_Error</i>	LESS THAN	Null	THEN	<i>Throttle</i>	Down
R3	IF	<i>Acceleration</i>	GREATER THAN	Null	THEN	<i>Throttle</i>	Up
R4	IF	<i>Acceleration</i>	LESS THAN	Null	THEN	<i>Throttle</i>	Down

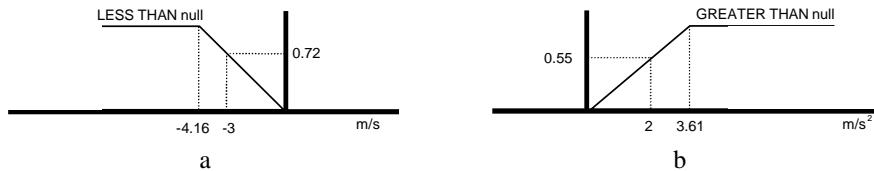
The *Speed\_Error* is similar to a classical control proportional component, and *Acceleration* is the derivative component. This means that, when the speed of the car is not at the target value, the aim of the *Speed\_Error* is to adjust the throttle pressure command, and the aim of *Acceleration* is to smooth out the actuation of the output.

Let us give a short overview of how a fuzzy control system can be implemented so that computationally cryptic statements such as the above can yield a number to feed a digital-to-analog converter.

In our example, we will assume that the instantaneous crisp values of the input variables are:

$$\text{Speed error} = -3 \text{ m/s}; \text{acceleration} = 2 \text{ m/s}^2$$

Figure 9.4a and b shows the result of the fuzzification of both crisp values. In this case, the degree of truth of the fuzzy variable *Speed\_Error* shown in Figure 9.4a for this input with the “LESS THAN *null*” label is 0.72. Similarly, Figure 9.4b shows that the degree up of *Acceleration* with “GREATER THAN *null*” is 0.55 for that crisp value.



**Figure 9.4.** Example of input variables fuzzification

The fuzzification with the other two linguistic labels for these input values have a truth degree of 0. Then, only the rules R2 and R3 are to be applied, with a weight of 0.72 for the first rule and 0.55 for the second.

Now, the controller applies the center of mass defuzzification method, and the output for the accelerator pedal will be Equation (9.1). The output fuzzy variable accelerator has two linguistic values “up” and “down”. Thus the output is computed in Equation (9.4).

$$x_{\text{out}} = \frac{0.72 \times \text{up} + 0.55 \times \text{down}}{0.72 + 0.55} = \frac{0.72 \times 1 + 0.55 \times (-1)}{0.72 + 0.55} = 0.11 \quad (9.4)$$

This means that the fuzzy controller will send an increment command to the accelerator within the normalized interval from -1 to 1: 0.11 in this case.

#### 9.2.2.2 Fuzzy Adaptive Cruise Control Extension

This second control system is an extension of the above-mentioned fuzzy CC, with the objective of keeping a safe distance from the vehicle ahead in the same lane of the road. This operation involves the controller automatically adapting speed to the speed of the preceding vehicle by overriding the CC target speed maintenance. The main application of this controller is for highway driving and platoons to improve driver and passenger safety and comfort in a high-speed driving environment. The extreme situation is when the preceding car stops, in which the ACC equipped car must stop too. This is a classical scenario in traffic jam driving, and is generically named Stop&Go. It should be clear that this is not a different controller; it is an upgrade of the CC controller. If no other car is detected or the headway with a detected car is long enough, the ACC controller works like the CC controller.

Keeping a user-defined safe distance from the next vehicle is a speed-dependent function. This means that the inter-vehicle gap should be wider as the speed is higher. This is the foundation of the *time-headway* concept: a time-dependent safety distance from the preceding car. For example, if we set a safety time-gap of 2 *seconds*, the space-gap is 22.2 *meters* driving at 40 *km/h*, but would be about 55.5 *meters* if we were driving at a speed of 100 *km/h*. The time-gap

setting depends on the car's braking power, the weather, the maximum driving speed, *etc.*

Two new input fuzzy variables, a new rule and two rule modifications were added to the controller to perform ACC. The output is the same as for the CC controller: the actuation of the throttle pedal.

A new term is introduced in the definition of this controller and used in the associated experiments. The vehicle circulating in front of the ACC equipped car will be termed pursued. The pursuer car will be the automatically driven vehicle that adapts its speed to the preceding vehicle. At this point, let us define the new input variables:

- Current time gap ( $TG_{Current}$ ): this is the time headway, that is, the time it takes the pursuer vehicle to reach the pursued vehicle's current position at its present speed in seconds. The mathematical representation is

$$TG_{Current} = \frac{x_{Pursued} - x_{Pursuer}}{v_{Pursuer}} \quad (9.5)$$

where  $x_{Pursued}$  and  $x_{Pursuer}$  are the x coordinate of the pursued and the pursuer cars over the reference line, in meters, and  $v_{Pursuer}$  is the speed of the pursuer car in meters per second.

- Target time gap ( $TG_{Target}$ ): this is the time-headway that the ACC system is to keep from the preceding vehicle set by the user. In commercial vehicles it should be between 1 and 2 *seconds*.
- Time-gap error ( $TG_{Error}$ ): this represents the time-gap error, that is, the difference between the user-defined target time-gap and the current time-gap, measured in seconds

$$TG_{Error} = TG_{Current} - TG_{Target} \quad (9.6)$$

- Time-gap derivative ( $TG_{Derivative}$ ): this is the variation of the current time gap with time. The equation used to calculate this variable for the control iteration  $i$  is

$$TG_{Derivative_i} = \frac{TG_{Current_i} - TG_{Current_{i-4}}}{4\Delta t} \quad (9.7)$$

A new sensor has to be added to gather the information on the preceding car's location. The control level is indifferent to the data source, because the system will work correctly with any sensor that supplies the required position, such as a laser, radar or a wireless local area network (WLAN) link, which is what was used in our case [36]. All the cars involved in our experiments are equipped with a real-time kinematics differential global positioning system (RTK-DGPS) and a WLAN card, and their positions are broadcast in a continuous loop so that every car knows at

every moment where the other one is. The new controller architecture schema is shown in Figure 9.5.

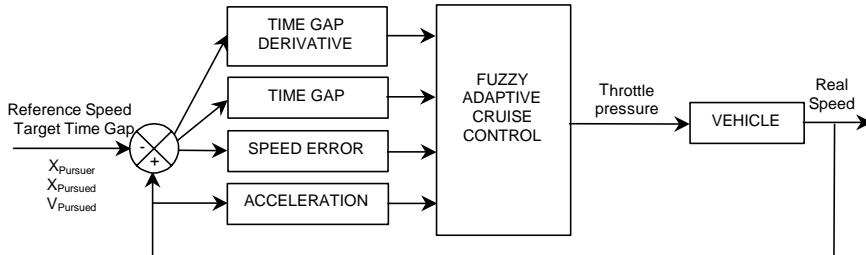


Figure 9.5. Fuzzy adaptive cruise control architecture

Two new input fuzzy variables are added to the CC controller in order to implement the ACC fuzzy controller: *Time\_Gap\_Error* and *d\_Time\_Gap*.

Two linguistic labels have been defined for the first variable: *near* and *far*. A diagram of the labels for this membership function is shown in Figure 9.2c.

*d\_Time\_Gap* has only one associated linguistic label called *negative*, and its units are headway-seconds per second. This is the variation of the time-gap per time unit. Its membership function is shown in the diagram in Figure 9.2d.

The last components of the ACC controller are the fuzzy rules. We added a new rule, and we also modified two other CC rules. The new set is shown in Table 9.2.

Table 9.2. Fuzzy control rules of the ACC controller

		Input fuzzy variables	Linguistic modifiers	Linguistic labels		Output fuzzy variable	Linguistic labels
R5	IF	<i>Speed_Error</i>	GREATER THAN	Null	THEN	<i>Throttle</i>	Up
R6	IF	<i>Speed_Error</i>	LESS THAN	Null			
	AND	<i>Time_Gap_Error</i>	GREATER THAN	Near	THEN	<i>Throttle</i>	Down
R7	IF	<i>Acceleration</i>	GREATER THAN	Null	THEN	<i>Throttle</i>	Up
R8	IF	<i>Acceleration</i>	LESS THAN	Null			
	AND	<i>Time_Gap_Error</i>		Far	THEN	<i>Throttle</i>	Down
R9	IF	<i>Time_Gap_Error</i>		Near			
	AND	<i>d_Time_Gap</i>		Negative	THEN	<i>Accelerator</i>	Up

The aim of these rules is to maintain the cruise control and to keep a safe distance. To do this, the gap control is only activated when the pursuer car is near to the pursued vehicle and is inhibited otherwise, with control switching to classical CC. Another significant point is that speed adaptation will only be activated when the pursued car is moving slower than the pursuer vehicle. Consequently, the acceleration rules (R6, and R8) need to be inhibited and the braking rules (R9) boosted. Let us look at the modified acceleration rules in detail.

R6: This rule forces the throttle signal to decrease when the pursuer car is near to the pursued vehicle so it will never accelerate as much as to collide with the other car.

R8: This rule only allows pressure to be applied to the throttle when the pursuer car is far away from the pursued vehicle.

R9: This rule causes the controller to release the pressure on the throttle, as the pursuer vehicle approaches the safety distance limit. The derivative is included in this rule for system stabilization reasons.

The output variable of the controller is the same as in the CC.

### 9.3 Experiment Results

Once the above defined controller had been installed in the testbed cars, some experiments were run to demonstrate its performance. The first showed the behavior of the fuzzy CC controller, automatically tracking a route in which the target speed reference changes continuously. The second experiment shows fuzzy ACC controller performance when a car is circulating in front of the unmanned vehicle in the same lane and direction and at a variable speed, lower than the vehicle's own target speed.

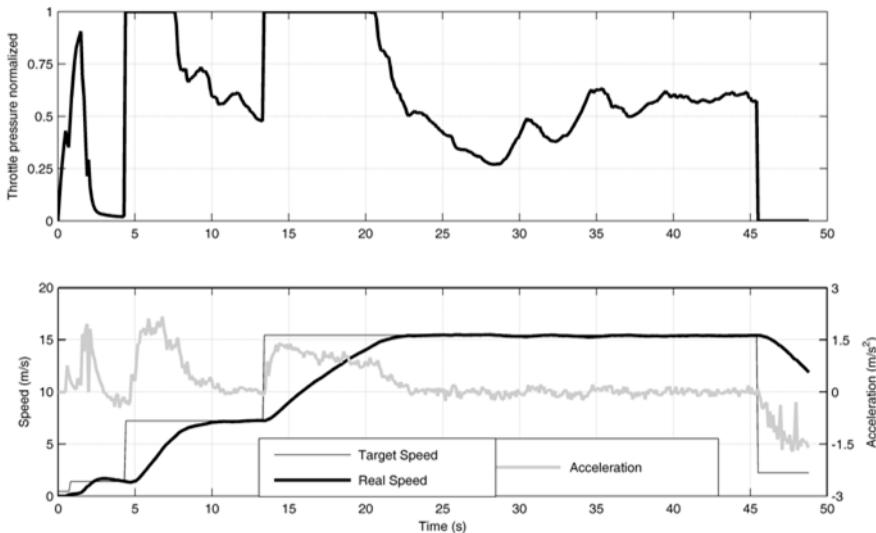
Figure 9.2 shows the results of the first lateral control test. This figure is composed of two graphs, the bottom graph containing the input variables, target speed, real speed and acceleration. The top graph represents the controller output, that is, the normalized voltage signal that emulates the throttle and is directly sent to the internal car's onboard computer.

This experiment illustrates how a target speed that varies in time depending on the track events is reached and maintained. Figure 9.2 shows that the user has selected three different reference speeds: 2, 7.5, and 15 *meters per second*. The circuit is short, about 250 *meters*, and, therefore, the experiments only go up to 15 *m/s*.

In this experiment, the fuzzy cruise control performance is very smooth, with a negligible error. This way, once the car has reached its reference speed, the stationary average error is 0.052 *m/s*, with a variance of 0.036 *m/s*. One important cruise control system evaluation measurement is how comfortable and safe passengers feel. This is critical for automatic systems applied to mass-produced cars, because car buyers are usually not very fond of such tasks being automated.

The second set of experiments consists of combining the safety headway maintenance and stop-and-go capability: the fuzzy ACC system. In this case, both testbed cars are equipped with a computer, an RTK-GPS receptor and WLAN, but the ACC is only activated in the pursuer vehicle. The pursued car is manually

driven to simulate real conditions in which the reactions of the car ahead are unpredictable. The pursuer car must adapt its speed to all driving situations, both to a car circulating in the same direction and to being alone on the road. In the first case, the rear car's speed must be adapted to maintain a user-defined safety gap up until it reaches the target cruising speed also defined by the driver. The CC described above will be in effect when the car is alone on the road.



**Figure 9.6.** Step reference sequence system response

During the 90 seconds that this experiment lasts, the pursued car runs at a speed ranging from 0, when the car is stopped, to 3 m/s. The pursuer car's target speed is always higher than the pursued vehicle's. Therefore, the cruise control will only maintain this speed when the pursuer is farther away than the pre-selected safety gap. We have also pre-set a 2-second target safety headway for the pursuer car. This means that if the pursuer's target speed is higher than the pursued vehicle's, the pursuer's advanced cruise control will keep a safety distance of 2 seconds from this preceding car. The reason for selecting this value is that 2 seconds are enough to meet the safety requirements in a limited environment, like our circuit.

This experiment is illustrated in Figure 9.7. This figure is made up of four different graphs. The bottom graph shows the real speed of both cars for the duration of the experiment. The third graph is the real-time headway time-gap between these testbed cars. The second graph represents the inter-vehicle distance, in meters, including the length of the pursued car (4 meters). The top graph shows the pressure on the throttle of the pursued car, where 0 means that the pedal is fully released and 1 that the throttle fully depressed.

At the beginning of the experiment neither car is moving, and there is a distance of about 16 meters between the two. The driver of the pursued car starts slowly while the automatic driver of the pursued car sets the target speed at 4 m/s. The time gap is initially very high, because the speeds are so low. As the pursuer

car speed increases, the gap narrows. After the first 10 *seconds*, the gap narrows drastically to about 2 *seconds*.

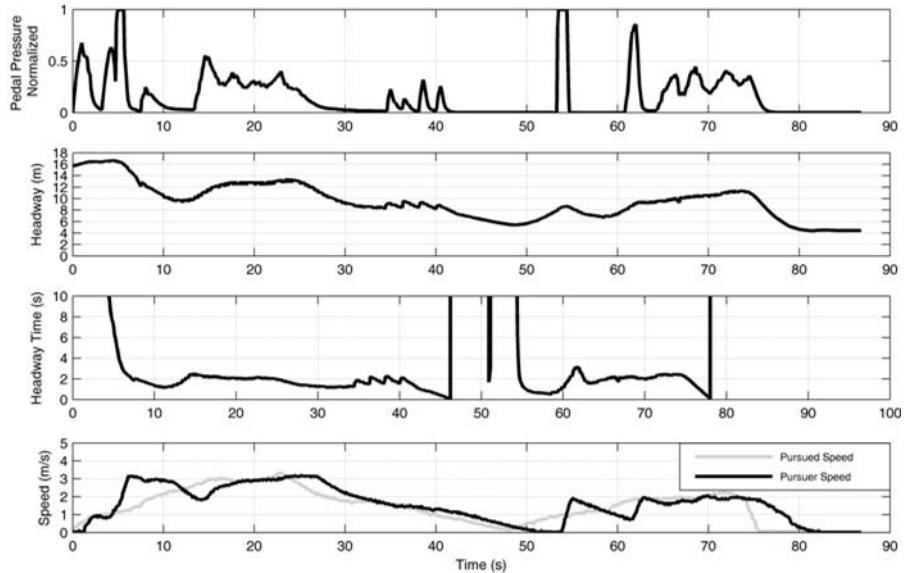


Figure 9.7. ACC test

After about 50 *seconds*, the pursued car stops. Here, the pursuer car approaches the other car until there is a gap of about 2 *meters* (6 in the graph), when it stops too. The reason for this change of units is that when the pursuer speed tends to zero, the time-gap tends to infinite and is not, in this case, useful for control purposes because the cars would crash. This is illustrated in the headway time graph at around the 50th *second*. The choice of 2 *meters* is very reliable, because the pursuer is slowing down and is about to stop when the pursuer reaches this distance.

After this first stop, the pursued car accelerates sharply. The pursuer vehicle accelerates too. The figure shows that whereas the time gap is about 2 *seconds*, the distance in meters increases because the speed of the two cars increases rapidly. Generally, the average gap interval ranges from 3 to 1.5 *seconds*.

The headway time graph also shows two reductions in the time gap at 40-50 and 70-80 *seconds*. The reason for this decrease is that there is a delay in pursuer braking after the pursued car brakes hard. In both cases, the controller stops the second car easily, because the speed is very low. The distance between the cars is never less than 2 *meters*. To improve safety at such low speeds, we recommend increasing the minimum safety gap to 3 or 4 *seconds*.

As mentioned in the introduction, [1, 37] and the documentation of different commercial ACCs, they work only at speeds above 30 *km/h*, which rules out their use in slow traffic and traffic jams. The ACC described in this chapter behaves in a

similar way at these speeds, but it does also work at very low speeds and performs stop and go.

## 9.4 Conclusion

This study shows that powerful controllers can be built using global navigation satellite systems (GNSS) for data input. It also demonstrates that, when the right techniques are used, these controllers outperform classical systems and provide a different point of view for implementing intelligent transportation systems.

Furthermore, it illustrates that the combination of ACC+Stop&Go relieves human drivers trapped in traffic jams of tedious tasks, thus improving driving safety.

In our opinion, fully automated driving is utopian and will not be achieved for many years. However, the experiments described here are a step in this direction. Our applications belong to the field of driving safety components and can be classed as driving aids.

## Acknowledgement

This work has been funded by the projects: Ministry of Education's ISAAC, "System Integration for Active Driving Assistance", and Ministry of Public Works' COPOS, "Navigation Systems Cooperation".

## References

- [1] Willie D. Jones, "Keeping Cars from Crashing," IEEE Spectrum, September 2001, pp 40 – 45.
- [2] Jesse Crosse, "Tomorrow's World," Automotive World, January/February 2000, pp 46 – 48.
- [3] STARDUST, "Scenarios and Evaluation Framework for City Case Studies", European Commission Fifth Framework Programme Energy, Environment and Sustainable Development Programme Key Action 4: City of Tomorrow and Cultural Heritage, Deliverable 2, 3, 2002.
- [4] Colleen Serafin, "Driver Preferences and Usability of Adjustable Distance Controls for an Adaptive Cruise Control (ACC) System," Ford Motor Company Systems Technology Inc. October 1996.
- [5] P. A. Ioannou and C. C. Chien, "Autonomous Intelligent Cruise Control", IEEE Transactions on Vehicular Technology, pp 657-672, Vol. 42, Nov 1993.
- [6] R. Holze, P. Protzel, J. Bernasch, K. Naab, "Adaptive Fuzzy Control for Driver Assistance in Car-Following", Proceedings of the 3rd European Congress on Intelligent Techniques and Soft Computing – EUFIT'95, Aachen, Germany, pp 1149-1153, Aug. 1995.
- [7] "Future-Oriented Technologies Revolutionize Active Vehicle Safety. Driver-assistance systems of the future", Volkswagen of America News, 20 January 2004.

- [8] J. I. Suárez, B. M. Vinagre, Y. Q. Chen, "Spatial Path Tracking of an Autonomous Industrial Vehicle Using Fractional Order Controllers", ICAR 2003 Congress, Portugal, 2003.
- [9] E. J. Rossetter, J. C. Gerdes, "Performance Guarantees for Hazard Based Lateral Vehicle Control", proceedings of the 2002 IMECE Conference, 2002.
- [10] S. Bentalva et al., "Fuzzy Path Tracking Control of a Vehicle", IEEE International Conference on Intelligent Vehicles, pp 195-200, 1998.
- [11] S. Sheikholeslam and C. A. Desoer, "Design of Decentralized Adaptive Controllers for a Class of Interconnected Nonlinear Dynamical Systems: Part I", PATH Technical Memorandum 92-1, Department of Electrical Engineering and Computer Sciences, Institute of Transportation Studies University of California, Berkeley, February 3, 1992.
- [12] R. Holve, P. Protzel, K. Naab, "Generating Fuzzy Rules for the Acceleration Control of an Adaptive Cruise Control System", Fuzzy Information Processing Society, 1996. NAFIPS. Biennal Conference of the North American, pp 451-455, 1996.
- [13] D. A. Pomerleau, "ALVINN: An Autonomous Land Vehicle In a Neural Network", Advances in Neural Information Processing Systems 1, Morgan Kaufmann, 1989.
- [14] M. Sugeno, "An Introductory Survey to Fuzzy Control," Information Sciences 36, 1985.
- [15] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modelling and Control", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-15, No. 1, pp 116-132, January/February, 1985.
- [16] L. A. Zadeh, "Fuzzy Sets", Information and Control 8, pp. 338-353, 1965.
- [17] E. H. Mamdani, "Application of Fuzzy Algorithms for Control of a Simple Dynamic Plant", Proc. IEE, 121, 12, pp 1585-1588, 1974.
- [18] J. E. Naranjo, Carlos González, Ricardo García, Teresa de Pedro, and Rodolfo E. Haber "Power Steering Control Architecture for Automatic Driving", IEEE Transactions on Intelligent Transportation Systems, Vol. 6, No. 4, pp 406-415, December 2005.
- [19] M. Sugeno, I. Hirano, S. Nakamura and S. Kotsu, "Development of an Intelligent Unmanned Helicopter", IEEE International Conference on Fuzzy Systems, Vol. 5, pp 33-4, 1995.
- [20] M. Sugeno, "Industrial Applications of Fuzzy Control", North Holland, New York, 1985.
- [21] M. Persson, F. Botling, E. Hesslow, R. Johansson, "Stop&Go Controller for Adaptive Cruise Control", Control Applications, 1999, Proceedings of the 1999 IEEE Conference on, Vol. 2, pp 1692-1697, 1999.
- [22] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, H. Fujiri, "Vehicle Control Algorithms for Cooperative Driving With Automated Vehicles and Intervehicle Communications", IEEE Transactions on Intelligent Transportation Systems, Vol. 3, No. 3, September 2002, pp 155 – 161.
- [23] Q. Chen, U. Ozguner, K. Redmill, "Ohio State University at the DARPA Grand Challenge: Developing a Completely Autonomous Vehicle", IEEE Intelligent Systems, Sep-Oct 2004, Vol. 19, No. 5, pp 8-11.
- [24] Mark Bursa. "Big names invest heavily in advanced ITS technology," ISATA Magazine December/January 2000, pp 24 – 30.
- [25] JM. Blosseville, M. Parent. "The French Program: La Route Automatisée," IEEE Intelligent Systems, May/June 2000, pp 10 – 13.
- [26] California PATH Program. "Literature Review of ADAS/AVG (AVCSS) for Japan and United States," June 27, 2001.
- [27] R. Bishop, "Intelligent Vehicle Applications Worldwide," IEEE Intelligent Systems, January/February, 2000, pp 78 – 81.

- [28] L. A. Zadeh, "Fuzzy Languages and their Relation to Human and Machine Intelligence", *Man and Computer Proc. Int. Conf.*, Bordeaux, pp 130-165, 1970.
- [29] L. A. Zadeh, "Approximate Reasoning Based on Fuzzy Logic", Memorandum No. UCB/ERL M79/32, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, May 1979.
- [30] L. A. Zadeh, "A Theory of Approximate Reasoning", *Machine Intelligence 9*, J. E. Hayes, D. Michie and L. I. Mikulich, editors, Wiley, 1979.
- [31] Lofti A. Zadeh, "A New Direction in AI, Towards a Computational Theory of Perceptions," *AI Magazine*. Spring 2001, pp 73 – 84.
- [32] R. Garcia and T. De Pedro, "Modeling a Fuzzy Coprocessor and Its Programming Language," *Mathware and Soft Computing Vol. V*, No. 2-3, 1998, pp 167 – 174
- [33] R. Garcia, T. De Pedro, J. E. Naranjo, J. Reviejo and C. Gonzalez, "Frontal and Lateral Control for Unmanned Vehicles in Urban Tracks," *IEEE Intelligent Vehicle Symposium*, June, 2002.
- [34] A.V. Patel, B.M. Mohan, "Some Numerical Aspect of Center of Area Defuzzification Method", *Fuzzy Sets and Systems*, 132, pp 401, 409, 2002.
- [35] Radix, J.-C. "Introduction au Filtrage Numérique". Eyrolles, Paris, 1970.
- [36] JE Naranjo, "Sistemas de Comunicaciones Inalámbricos Aplicados a la Conducción Automática de Vehículos," *Degree Thesis Universidad Politécnica de Madrid Spain*, September 2001.
- [37] "Evaluation of the Intelligent Cruise Control System. Vol. I – Study Results", Research and Special Programs Administration, Volpe National Transportation Systems Center, Cambridge, MA 02142-1093.

## Fuzzy Logic Control for Automobiles I: Knowledge-based Gear-position Decision

Guihe Qin, Anlin Ge, and Ju-Jang Lee

### 10.1 Introduction

In a road vehicle, the functions of transmission are to match the running state of the engine to the motion states of the vehicle. The aims are to optimize the fuel economy and the dynamic features, keep the vehicle safe and controllable, and make passengers more comfortable. The ratio of transmission, which is determined with the gear positions, is changed according to engine states, vehicle states, and road situations. Transmissions are classified as Automatic Transmissions (AT) and Manual Transmissions (MT). A traditional AT is composed of a planetary gearbox and a torque converter. Shifting automatically, it is convenient to drive an AT vehicle. However, AT's transfer efficiency is low, and it is more difficult to manufacture and maintain than MT. MT is composed of a gearbox and a dry clutch. It has to be operated manually. However, it has the advantages of high efficiency and is easy to manufacture and maintain. Automated Manual Transmission (AMT) achieves automatic shifting function by electronic control of MT. AMT has the advantages of both MT and AT [1, 2]. To implement automatic shifting AMT must have the ability of automatic gear ratio selection, *i.e.*, gear position decision (GPD), and the ability of automatic shifting control. Shifting control is a typical mechanical device control problem. Much research has been carried out to improve its performance [3-5]. GPD involves not only vehicle parameters but also driver's intentions and road environments. In this chapter, we focus the discussion on the GPD problem only.

GPD is one of the key technologies of AMT. Strongly affecting the performances of the AMT vehicle, GPD has been a most important research topic since AMT appeared in the 1980s [6, 7]. With early AMT technology, GPD methods were mainly adopted from AT. The main tasks of the technology at the time were to implement basic automatic shifting functions. The traditional GPD methods of AMT, the same as those used in AT, are usually based only on such vehicle state parameters as speed, throttle opening degree and acceleration [8, 9]. The GPD methods are very sensitive to the parameters and can't adapt to varying road environments. They also can't cater for a driver's operating intentions [10, 11]. Because AMT is not as flexible as AT in mechanical characteristics, the

defects derived from the drawbacks are more serious [8]. In complex road environments, the drawbacks cause AMT to carry out many unnecessary gear changes and not select proper gear positions. These may reduce the average driving force and the fuel economy because of interruption of power in shifting, and decrease the life of mechanical parts because of extra wear. Ignorance of the driver's intentions causes more operation of brake or accelerate. These problems are initially studied for ATs. With the development of the technology, some ideas have been presented from simple mechanical to software-based approaches. Among them are knowledge and fuzzy logic based tactics. Weil [11], Bastian [12], and Sakaguchi [13] proposed the idea of introducing driving knowledge into shifting schedules of AT, and respectively researched the "wind road", "slop road", and "load adaptation" situations. Graf [14], Shimizu [15] and Hayashi[16] presented neuro-fuzzy based GPD concepts to enable the control algorithms to adapt to load variance and different type drivers. As new power trains are introduced and superpower micro-controllers are used in vehicles, more advanced control algorithms for automatic transmission systems are presented [17]. Most of the work focuses on one or several problems and conceptualizing research. Little work has been carried out general intelligent GPD models, especially the work on AMT and practical methods.

In this chapter, we present in detail a general knowledge-based GPD model for AMT and its realization. The driver's intentions as well as road environments are introduced to improve the performances of AMT. It employs the information detected with in-vehicle sensors. The principle aim in developing the algorithm is to improve the performances of AMT as much as possible with no or little extra hardware additions. The basic idea is to use fully the available information and mimic the GPD course of experienced drivers. The model is based on driving knowledge. A three-layer intelligent control structure of AMT is achieved from the analysis of an experienced driver's shifting operation. The topic mainly focused on is the top layer, *i.e.*, GPD. The test results show that, compared with the traditional two parameters GPD method [8], the presented method makes the test car use less unnecessary shifting, select gear position better and obtain higher subjective assessment scores.

The rest of the chapter consists of three sections. Section 2 covers the formulation of GPD and Section 3 illustrates experimental results. Section 4 provides the conclusion and outlines future work.

## 10.2 Knowledge-based GPD (KGPD)

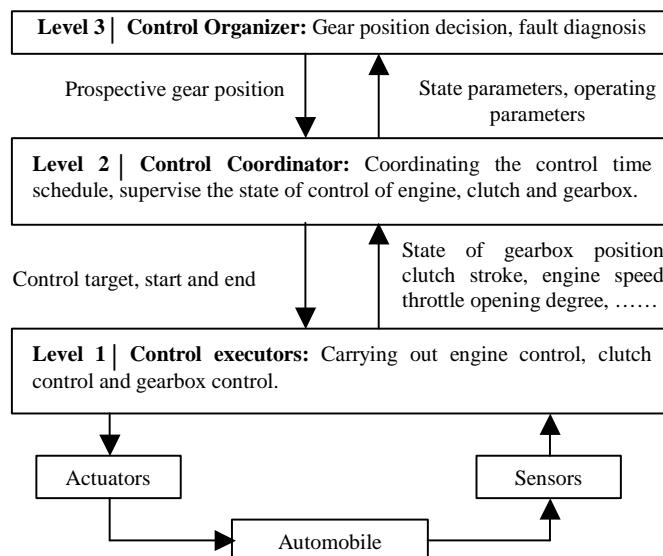
### 10.2.1 Basic Ideas

GPD is a very complex course. Essentially, it's an optimizing problem. Under different road environments and driving intentions the optimum gear position is selected depending on different principles. It is very difficult to model the course with pure mathematical functions. On the other hand, an experienced driver can select the gear positions properly in any driving situation when operating manual

transmission vehicles. Inspection of the driving process can provide useful ideas to design a more practical AMT control system. From the consultation of experienced drivers and vehicle experts as well as the driving handbook [18], we figure out the course of shifting as follows:

- 1) Generating intentions of the interval based on the travel purpose, the sensed road environments and the vehicle states.
- 2) Making a GPD; if the present (GP) is not the same as the calculated one, shifting will be carried out.
- 3) Figuring out the shifting operation-time sequences of the accelerator, clutch stroke and the gear-operating lever.
- 4) Carrying out the operation of the accelerator, clutch stroke and the gear-operating lever respectively according to the time sequences.

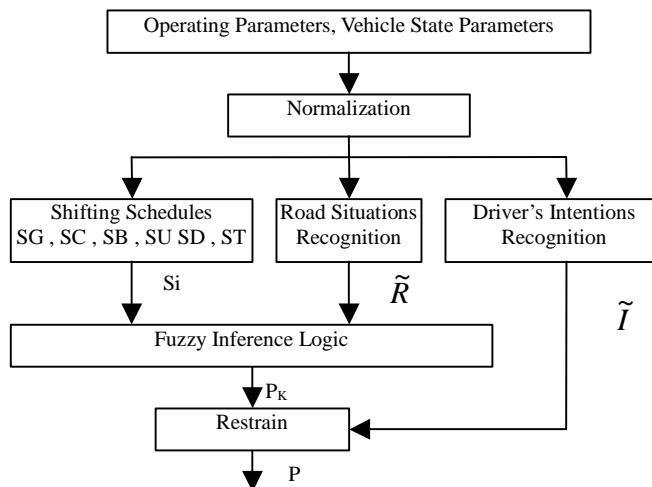
To mimic the course, the control system of the AMT is adapted to a three-layer intelligent-control system structure [19] as shown in Figure 10.1. This is the basic structure of the AMT control system. The top layer finishes the gear position decision, the middle layer coordinates the engine control, the clutch control and the gearbox control, and the bottom layer carries out the control operations of engine (throttle opening), clutch stroke and gear box respectively. The functions of the bottom two layers are shifting control operations. In the AMT, the throttle opening is controlled with a stepper motor, the clutch with a cylinder, and the gearbox with two cross cylinders. More details of the parts can be found in [2, 20]. In this chapter, the top layer, *i.e.*, GPD algorithm is to be explained in the next sections.



**Figure 10.1.** AMT intelligent control architecture

The experienced driver's GPD can be summarized as follows:

- 1) Sensing road environments and estimating vehicle states.
- 2) Figuring out the features of the road environments and adjusting the vehicle according to driving knowledge and experiences.
- 3) Recalling gear-selection rules suitable for the road situations and intentions.
- 4) Selecting a gear position according to operation rules and experiences.
- 5) Carrying out operation of shifting to the selected gear position if it is not same as the present one.



**Figure 10.2.** General structure of the knowledge based gear position decision algorithm

This course is modeled as Figure 10.2, which is a KGPD algorithm that can synthetically use the information of road environment and driver's intentions as well as the automotive state parameters as an experienced driver behaves. The algorithm is composed of a road-environment estimator (E1), a driver's intention estimator (E2), a shift-schedule knowledge base (S) suited for every typical situations, a fuzzy-inference algorithm (I) to calculate the most suitable gear position, and the measure and normalization part to measure and normalize the vehicle state parameters as well as the driver's operating parameters. The normalized data are sent to the estimators and the shift schedule algorithms. The estimators E1 and E2 calculate the present road environments ( $\tilde{R}$ ) and the driver's intentions ( $\tilde{I}$ ) (refer to Section 10.2.2). The estimated result ( $\tilde{R}$ ) is used to select the gear position suitable for the road situation under the restraint of  $\tilde{I}$ . The basic meaning of the algorithm is “if the road environment is  $\tilde{R}$  and the intention is  $\tilde{I}$ , then use schedule S( $\tilde{R}$ ) to calculate the optimum gear position with restriction of  $\tilde{I}$ ”.

The basic idea of the method is that the optimal schedules for any situations are stored in the knowledge base, and that the one used to calculate the gear position is selected by the dynamically recognized situation. Every part of the model is to be depicted in the rest of the section.

### 10.2.2 Parameter Measurement and Normalization

The normalized measured parameters include the following:

1) *Throttle opening  $\alpha(i)$  degree* ( $\alpha(i)$  means the normalized throttle opening degree at  $t_i$ . “ $i$ ” denotes the sampling time  $t_i$ . In the rest of the chapter the meaning of  $(i)$  is same as in  $\alpha(i)$ ): detected with a potentiometer mounted to the throttle. The throttle-opening degree varies from 0% to 100% corresponding  $\alpha(i)$  from 0 to 1.

2) *Vehicle velocity  $v(i)$* : derived from speed of the output shaft of the transmission. The speed is measured with an encoder. The relation between real vehicle speed and  $v(i)$  is

$$v(i) = \begin{cases} V(i)/(80 \text{ km/h}); & V(i) \leq 80 \text{ km/h} \\ 1; & V(i) > 80 \text{ km/h} \end{cases} \quad (10.1)$$

where  $v(i)$  is the directedly measured velocity. When  $v(i)$  is bigger than 80 km/h the gear position is at the highest for any situations in the tested car. So  $v(i)$  is normalized with 80 km/h.

3) *Variation of throttle opening degree  $\alpha'(i)$* : calculated from  $\alpha(i)$ :

$$\alpha'(i) = \begin{cases} (\alpha(i) - \alpha(i-2) + 1)/10 & ; |\alpha(i) - \alpha(i-2)| \leq 5 \\ 1 & ; |\alpha(i) - \alpha(i-2)| > 5 \end{cases} \quad (10.2)$$

4) *Variation of vehicle velocity  $v'(i)$* : calculated from  $v(i)$ :

$$v'(i) = (v(i) - v(i-5) + 1)/2 \quad (10.3)$$

5) *Braking frequency  $B_f(i)$* : defined as the change times of the braking state in a time interval. To calculate  $B_f(i)$  a variable  $B(i)$  is introduced to count the breaking action. At the sampling point, the breaking state is inspected. If it is different from the last one,  $B(i)$  is increased by 1:

$$B_f(i) = \begin{cases} (B(i) - B(i-50))/5 & ; B(i) - B(i-50) \leq 5 \\ 1 & ; B(i) - B(i-50) > 5 \end{cases} \quad (10.4)$$

6) Frequency of  $\alpha'(i)$  passing zero  $\alpha_f(i)$ : defined as the times of  $\alpha'(i)$  changing sign in a time interval and is the same as the method to compute  $B_f(i)$ ; to calculate  $\alpha_f(i)$  a variable  $F(i)$  is introduced to count the sign change of  $\alpha'(i)$ . At the sampling point the sign of  $\alpha'(i)$  is inspected. If it is different from the last one,  $F(i)$  is increased by 1:

$$\alpha_f(i) = \begin{cases} (F(i) - F(i-30))/3 & ; F(i) - F(i-30) \leq 3 \\ 1 & ; F(i) - F(i-30) \mid 3 \end{cases} \quad (10.5)$$

7) *Accumulation of braking time*  $B_t(i)$ : defined as the accumulated time of breaking state in a interval. To obtain  $B_t(i)$  a clock is used to measure the duration of each braking. Whenever the break pedal is stamped down the clock is reset. When the pedal is released the clock is stopped and the recorded time is added to the variable  $C_t(i)$ . At the sampling point, if the clock is counting, the counted time is added to  $C_t(i)$  and then the clock is reset:

$$B_t(i) = \begin{cases} (C_t(i) - C_t(i-20))/2/20 & ; C_t(i) - C_t(i-20) \leq 40 \\ 1 & ; C_t(i) - C_t(i-20) \mid 40 \end{cases} \quad (10.6)$$

The parameters are selected because they can be measured with existing sensors and are sensitive to the road environments. The combination of  $\alpha(i)$ ,  $v(i)$  and  $v'(i)$  reflect the vehicle dynamic features including road resistance at and before the moment.  $\alpha'(i)$ ,  $B_f(i)$ ,  $\alpha_f(i)$ , and  $B_t(i)$  are sensitive to what the driver senses and what the driver wants [21].

### 10.2.3 Road Environment and Driver's Intention Recognition

According to the need of GPD [20], road environments are divided into the typical patterns, as follows :

- “Good road” (G), *i.e.*, flat straight and spacious
- “Complex road” (C), *i.e.*, jam or narrow or winding
- “Bump road” (B)
- “Sharp up road” (U)
- and “Sharp down road” (D)

To recognize them, we consider them as patterns defined by space  $(\alpha, v, \alpha', v', \alpha_t, B_f, B_t)$ . The pattern set is

$$\psi = \{G, C, B, U, D\} \quad (10.7)$$

The pattern vector is defined with the measured parameters as

$$X_j = (\alpha_j, v_j, \alpha'_j, v'_j, \alpha_{ij}, B_{fj}, B_{tj}) \quad j \in \psi \quad (10.8)$$

Unknown vector at time  $t_i$  is

$$X(i) = (\alpha(i), v(i), \alpha'(i), v'(i), \alpha_i(i), B_f(i), B_t(i)) \quad (10.9)$$

For all the data of the parameters that are normalized (refer to Section 10.2.1), the pattern vector is of the form of a fuzzy set defined by the universe of discourse  $\{\alpha, v, \alpha', v', \alpha_i, B_f, B_t\}$ . With an experienced driver in the vehicle we sample the data in every situation. For situation  $j$ ,  $j \in \psi$ , a group of vectors are obtained. The center of the vectors is taken as  $X_j$ .

Defining a measure of distance of two vector points (Y and Z) in the space of  $(\alpha, v, \alpha', v', \alpha_i, B_f, B_t)$  as

$$\delta(y, z) = 1 - \frac{1}{\sum_{k \in p} \omega_k} \cdot \sum_{j \in p} |y(j) - z(j)| \cdot \omega_j \quad (10.10)$$

where  $p$  represents the set  $\{\alpha, v, \alpha', v', \alpha_i, B_f, B_t\}$ ,  $\omega_k \in [0,1]$  is the weight attached to the element  $k$  of vectors  $y$  and  $z$ . It is added to adjust the significance of the corresponding parameter. Evidently,  $\delta(y, z)$  is of the following properties:

- 1) Being a map of  $X \times X \rightarrow [0,1]$
- 2) If  $y = z$ ,  $\delta(y, z) = 1$
- 3)  $\delta(y, z) = \delta(z, y)$
- 4) If  $y \subset A \subset z$  or  $y \supset A \supset z$ ,  $\delta(y, A) \geq \delta(y, z)$

According to Wang and Li [22],  $\delta(y, z)$  defines a measure of near grade of two fuzzy sets. It can be used to describe the degree of difference of fuzzy sets. The value of  $\delta(X(i), X_j)$ , that is the near grade between  $X(i)$  and  $X_j$ , describes the grade that the present situation belongs to pattern  $j$ . Construct a set  $\tilde{R}$  as

$$\tilde{R} = \{\tilde{R}(j)\} \quad ; \quad \tilde{R}(j) = \delta(X(i), X_j), \quad j \in \psi \quad (10.11)$$

For  $\delta(X(i), X_j) \in [0,1]$ ,  $\tilde{R}$  is a fuzzy set defined on  $\psi$ , and its membership, that is the near degree between  $X(i)$  and  $X_j$ , describes how much the mode  $j$  belongs to the present situation. Therefore,  $\tilde{R}(j)$  provides the information of what mode the road now belongs to.

The driver's intentions are divided into three patterns: "acceleration" (A), "keeping the state" (K), and "slowing down" (S). A fuzzy set defined on

$\{A, K, S\}$  is employed to describe the patterns

$$\tilde{I}(i) = \left\{ \tilde{I}(j) \mid j \in \{A, K, S\} \right\} \quad (10.12)$$

where

$$\begin{aligned} \tilde{I}_i(A) &= \max((\alpha'(i) - 0.5), 0) / 0.5 \\ \tilde{I}_i(K) &= (1 - |\alpha'(i) - 0.5| / 0.5) \cap (1 - B_i(i)) \\ \tilde{I}_i(S) &= (1 - \alpha'(i)) \cup B_i(i) \end{aligned} \quad (10.13)$$

They are derived directly from the driving knowledge. The meaning of the definition is that if  $\alpha'(i)$  is bigger than 0.5 (throttle opening is increased) the driver want to accelerate, if  $\alpha'(i)$  is about at 0.5 (unchanged) and  $B_i(i)$  is small the driver wants to keep the speed, and if  $\alpha'(i)$  small or  $B_i(i)$  is big the driver wants to decrease the speed.

#### 10.2.4 Shift Schedules

Shift schedules are functions of vehicle state parameters used to calculate the optimum gear position. There are two kinds of methods to obtain the shift schedules. One is the optimization method which is to determine the shift points to optimise the fuel economy or dynamic performance. This is the traditional method to obtain the shifting schedules, and the method is relatively perfect. Here, the shift schedules  $S_G, S_U$ , which are respectively used at the situations G and U, are obtained with the method. The detail process to calculate the schedules can be found in [8]. The other method is to determine the shift points according to the operating rules and the experienced driver's knowledge. The shift schedules  $S_C, S_B, S_D, S_T$ , which are respectively used at the situations C, B, D and T, are obtained with the method as follows:

- 1) Listing all the operating rules in a situation
- 2) Sorting out all the control variants
- 3) Defining the fuzzy subsets of the variants
- 4) Inducing the fuzzy rules of GPD in the situation according 1) - 3)
- 5) Quantizing the control variants and calculating the shift-schedule tables

More details and the calculation of other schedules can be found in [20, 21].

All the shift schedules are expressed with datum tables and stored in the ECU of the AMT as a knowledge base. The output  $P_j$  of shift schedule  $S_j$  can be calculated with the corresponding table.

#### 10.2.5 Gear-Position Calculation

Based on the road environment information and intentions, with the shifting schedule knowledge, the GPD becomes such a problem as:

“If road situation  $R = j$ , then gear position  $P = P_j$ ;

The Road situation  $R = \tilde{R}(i)$ , the gear position  $P = ?$  (Crisp form).”

This is a fuzzy inference in which the rules are in crisp form. To correspondingly extend the membership of fuzzy set  $R = \tilde{R}(i)$  to the universe of discourse of  $\{P_j \mid j \in \psi\}$  we obtain the fuzzy-set gear position

$$\tilde{P}(i) = \{(\tilde{R}_j(i), P_j)\} \quad (10.14)$$

With the “First of Maxims” defuzzification method, the most possible road situation can be obtained as

$$P_k = \inf_{i \in \psi} \{P_j \mid j \in \text{Con}, \tilde{R}(j) = \max_{l \in \psi} (\tilde{R}(l))\} \quad (10.15)$$

Then,  $P_k$  is taken as the candidate gear position. At last the output gear position is determined with constraint of  $\tilde{I}(i)$  as

$$\text{If } \inf_{i \in \{A, K, S\}} \{i \in \text{Con} \mid I(i) = \max_{j \in \{A, K, S\}} (I(j))\} = A \cap P_0 \mid P_k,$$

then  $P = P_k$ ;  $P_0$  is the geared position;

$$\text{If } \inf_{i \in \{A, K, S\}} \{i \in \text{Con} \mid I(i) = \max_{j \in \{A, K, S\}} (I(j))\} = A \cap P_0 < P_k,$$

then  $P = P_k$  (with a delay);

$$\text{If } \inf_{i \in \{A, K, S\}} \{i \in \text{Con} \mid I(i) = \max_{j \in \{A, K, S\}} (I(j))\} = A \cap P_0 = P_k,$$

then  $P = P_0$ ;

$$\text{If } \inf_{i \in \{A, K, S\}} \{i \in \text{Con} \mid I(i) = \max_{j \in \{A, K, S\}} (I(j))\} = K,$$

then  $P = P_i$ ;

$$\text{If } \inf_{i \in \{A, K, S\}} \{i \in \text{Con} \mid I(i) = \max_{j \in \{A, K, S\}} (I(j))\} = S \cap P_0 \leq P_k,$$

then  $P = P_0$ ;

$$\text{If } \inf_{i \in \{A, K, S\}} \{i \in \text{Con} \mid I(i) = \max_{j \in \{A, K, S\}} (I(j))\} = S \cap P_0 \mid P_k,$$

then  $P = P_i$ . (10.16)

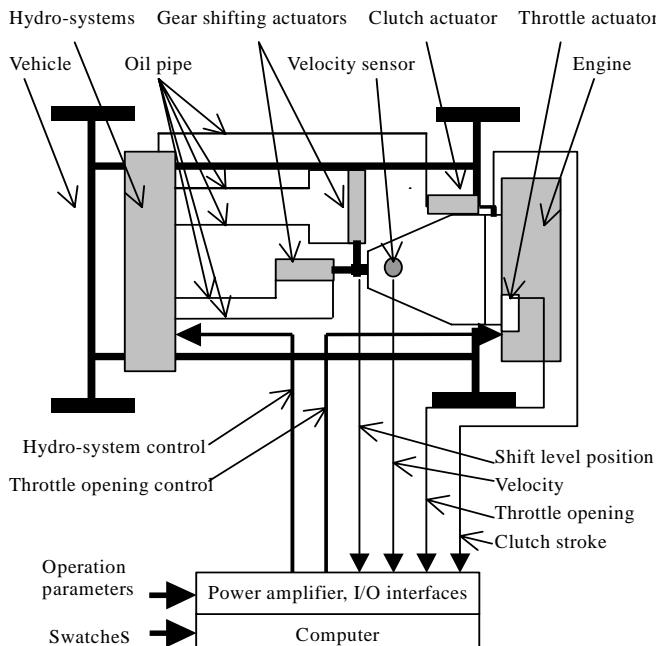
### 10.3 Experiment Test

The KGPD (Knowledge based Gear Position Decision) model was tested on the AMT mounted on a Shanghai Santana 2000 car with a five forward one backward gearbox transmission. The control was carried out with an in-car PC powered with UPS. The basic structure of the system is illustrated in Figure 10.3. The test was

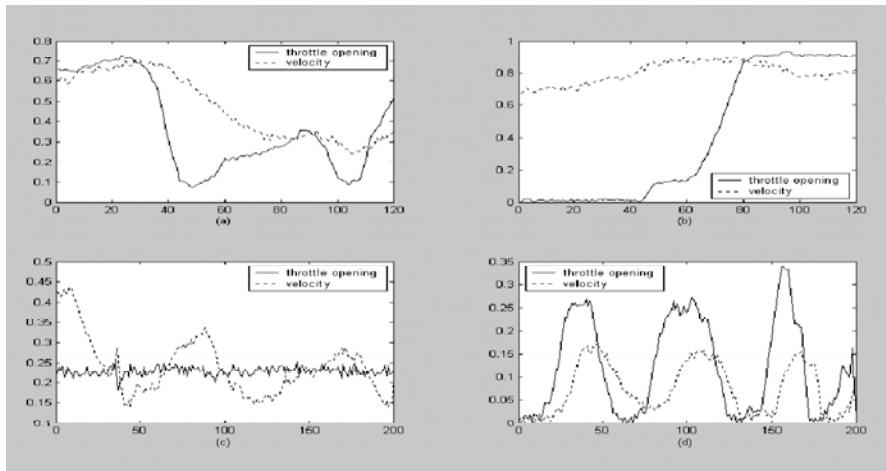
conducted in Changchun downtown and rural roads. The experiment was carried out as follows.

Ten drivers drove the car in turn. Every one drove four times on the same road (start from Jilin University to the rural area then back to the university). Among the four tests of every driver the KGPD was used twice, while the “dynamic three parameters schedule (DTP) “ was used twice more. The drivers were not informed of the difference. Every time the driver evaluated the shifting performance of the car. The results included subjective assessment of the drivers and the recorded data. Table 10.1 shows the results.

The AMT with the KGPD was evaluated higher than with DTP, and the average shifting times were evidently lower. From the recorded data we found that the decrease mainly appeared on “turn”, “bump”, “down and up” or “up and down” road as well as “deceleration” and “acceleration” situations. In other situations the shifting times with the two methods were about the same. Figure 10.4 illustrates the typical situations. In the situations with KGPD the AMT carried out no shifting. However, if DTP were used there would be an unnecessary shifting. In Figure 10.4a the AMT held a gear position 3 in this case. But if the DTP was used, the gear position would rise up to 4, then decrease to 3 according to the throttle opening and motion speed.



**Figure 10.3.** Experimental system



**Figure 10.4.** Test at different road situations

**Table 10.1.** Subjective assessment results

Drivers	Round1 (DTP)	Round2 (KGPD)	Round3 (DTP)	Round4 (KGPD)
1	75	37	85	34
2	80	40	85	36
3	75	38	75	36
4	70	35	75	34
5	75	38	80	37
6	80	39	90	36
7	85	36	90	36
8	85	40	90	37
9	80	38	80	35
10	70	37	85	36
<i>Average</i>	77.5	37.8	83.5	35.7
			77.0	37.6
			83.0	35.5

## 10.4 Conclusion

The road test data show that the AMT gives out less unnecessary shift, and the gear positions in "turn road", "sharp down", "deceleration", "bump road" are better than the one using common two parameter shift schedule in all situations. These prove

that the method improves the adaptation of AMT to the road. In the assessment the AMT obtained higher score than that with the traditional shift schedules. The subject assessment of functions is very important to cars. Generally, the road environment and intention information can be introduced into GPD and the performance of the AMT with the GPD are improved. The GPD model is general and practical.

Our future work is to focus on the self-adjusting ability of GPD, *i.e.* the ability to adapt to vehicle character change and different drivers.

## References

- [1] H.Yashimura, and A.Hirako, "Automated mechanical transmission control," in *Proc. of the International Congress on Transportation Electronics*, Dearborn, USA, 1986, pp 179-189.
- [2] Guihe Qin, Juxin Fan, Hongkun Zhang, and Anlin Ge, "Microcomputer control system of automated mechanical transmission," *Qiche Gongcheng/Chinese Automotive Engineering*, Vol. 21, No. 1, Jan. 1999, pp 21-25.
- [3] H.S.Jo, S.T.Jo, J.M.Lee, and Y.I.Park, "Analysis of the shift characteristics of a automated manual transmission in the parallel type hybrid drive train system of a transit bus," *Heavy Vehicle Systems*, Vol. 8, No. 1, 2001, pp 60-82.
- [4] J.Fredriksson, and B.Egardt, "Nonlinear control applied to gear shifting in automated manual transmissions," in *Proc. of the IEEE Conference on Decision and Control*, Vol. 1, Sydney, 2000, pp 444-449.
- [5] H.Tanaka, and H.Wada, "Fuzzy control of clutch engagement for automated manual transmission," *Vehicle System Dynamics*, Vol. 24, 4-5, 1995.
- [6] R.C.Holmes, R.R.Smyth, and D.Speranza, "Automation of mechanical transmissions", in *Proc. of the International congress on transportation electronics*, Dearborn, USA, 1984, pp 11-23.
- [7] R.C.Holmes, R.R.Smyth, and D.Speranza, "Manual transmission converted to automatic", *Automotive engineering*, Vol. 92, No. 1, 1984, pp 67-74.
- [8] Anlin Ge, *Vehicle Automatic Transmission Theory and Design*. Mechanical Industry Publishing House, Beijing, 1993, Chapter 4.
- [9] Anlin Ge, Jiaji Jiang, Wenzhi Wu, and Jianghang Chen, "Research on dynamic 3-parameter shift schedule of automatic transmission", in *Proc. of the 6th International Pacific Conference on Automotive Engineering*, Seoul, South Korea, 1991, pp 303-312.
- [10] H. Yamaguchi, etc., "Automatic Transmission Shift Schedule Control Using Fuzzy Logic," *SAE paper 930674*, 1993
- [11] H.G.Weil, G.Probst, and F.Graf, "Fuzzy expert system for automatic transmission control Control Applications," in *Proc. of 1st IEEE Conference on Fuzzy Systems*, Vol. 2, 1992, pp 716-721.
- [12] A.Bastian, S.Tano, T.Oyama, and T. Arnould, "FATE: fuzzy logic automatic transmission expert system," in *Proceedings of 1995 IEEE International Conference on Fuzzy Systems and the 2nd International Fuzzy Engineering Symposium*, Vol. 5, 1995, pp 5-6.
- [13] S.Sakaguchi, I.Sakai, and T.Haga, "Application of fuzzy logic to shift scheduling method for automatic transmission," in *Proc. 2nd IEEE International Conference on Fuzzy Systems*, Vol. 1, 1993, pp 52-58.
- [14] F.Graf, and W.Hauptmann, "Automotive Powertrain Control with Fuzzy Logic," *Chapter 6 of "Applications of Fuzzy Logic: Towards High Machine Intelligence*

- Quotient Systems*”, Prentice Hall PTR, Edited by M.Jamshidi, A.Titli, L.A.Zadeh and S.Boverie,1997, pp 109-129.
- [15] Y.Shimizu, S.D.Taniguti, Y.Kondo, A.Hayasi and K.Hirako, “Neuro-fuzzy Optimal Down Hill Driving Control for Automated Gear Shift System,” in *Proc. of the 1995 Artificial Neural Networks in Engineering, ANNIE'95*, St.Louis, USA, 1995, pp 563-568.
  - [16] K.Hayashi, Y.Shimizu, T.Yasuhiko and A.A.Hirako, “Neuro-fuzzy Transmission Control for Automobile with Variable Loads,” *IEEE Transactions on Control Systems Technology*, Vol. 3, No. 1, Mar, 1995, pp 49-52.
  - [17] Han-Sang Jo, Yeong-II Park, and Jang-Moo Lee, “Development of an advanced shift control algorithm for a hybrid vehicles with automated manual transmission,” *Heavy Vehicle Systems*, Vol. 7, No. 4, 2000.
  - [18] Yao Zhao, *Automotive Driver Handbook*. Renmin Transportation Publishing House, Beijing, 1989, Chapter 2.
  - [19] C.J.Harris, C.G.Moore, and M.Brown, *Intelligent Control : Aspects of Fuzzy Logic and neural Nets*. Woed Scientific Publishing Co. Pte. Ltd, Chapter 1.
  - [20] Guihe Qin, “Study on the Control Technology and Development of Control System of Automated Manual Transmission,” Ph.D. Dissertation, Jilin University of Technology,1997.
  - [21] Guihe Qin, Hongkun Zhang, Yulong Lei, and Anlin Ge, “Gear position decision method of automated mechanical transmission based on fuzzy inference,” *Chinese Journal of Mechanical Engineering (English Edition)*, Vol. 12, No. 1, 1999, pp 76-80
  - [22] Peizhuang Wang, and Hongxing Li, *Fuzzy System: Theory and Fuzzy Computer*, Science Publishing House, Beijing, 1996, Chapter 3.

## **Fuzzy Logic Control for Automobiles II: Navigation and Collision Avoidance System**

Andri Riid, Dmitri Pahhomov and Ennu Rüstern

### **11.1 Introduction**

In 1990, Nguyen and Widrow [1] applied a self-learning neural network to the truck backer-upper problem. The latter has become an acknowledged benchmark in non-linear control since then and numerous other techniques have been tried, including genetic programming [2] neurogenetics [3], simplified neural network solution through problem decomposition [4], *etc.*<sup>1</sup>. Computational overhead is usually very high in such applications, *e.g.* in [5] it is shown that about 20,000 back-up cycles are needed before a neural network learns and, even then, the backpropagation algorithm may not converge for some sets of training samples.

A simplified version of the control problem (consisting of the cab part only), on the other hand, has been heavily exploited in the field of fuzzy control [7-14]. Apparently, it seems to be one of these cases where the traditional application area of fuzzy logic - knowledge-based control - would be an appropriate solution. The ability to drive a car is a very common skill among people and thus it should not be too difficult to find an expert whose verbal instructions would then constitute the core of the control system. Car driving skill, however, is usually learned to a degree where it rarely intrudes on consciousness (the occasions when it does are unusual circumstances like a potential accident or a situation the driver is not used to (*i.e.* he/she has not yet learnt it)). Consequently, it is difficult to extract appropriate rules from the expert because of one's inability to explain how the action behind the steering wheel is exactly related to car positioning and further difficulties in putting it down in terms of fuzzy logic. The design of a knowledge-based controller therefore becomes much more difficult than was assumed in the first place. Though the computational load is low, controller design procedure is ill-defined and plagued with the curse of dimensionality that often leads to subpar performance.

---

<sup>1</sup> Perhaps the most interesting of these contributions is [6], where up to ten trailers are controlled representing those as Takagi-Sugeno models and applying linear matrix inequalities method.

Note that knowledge-based control would be very difficult to apply for the truck and trailer system because normal driving instincts cheat us when attempting to back up a trailer truck to a loading dock. The task is so difficult that a lot of practice is needed to master the skill. And even then, when a truck driver backs up toward a loading dock, he/she will go forward and backward numerous times in order to position the truck at the dock successfully. If the driver is not allowed to make forward movements, successful backing becomes improbable.

In [15] we have shown how the decomposition of the control problem can make controller design very natural and substantially simplify expert knowledge acquisition. In the decomposed view we focus on information concerning car optimal orientation in two-dimensional space (much easier to explain and understand than the minute actions on the steering wheel), which ultimately leads to the efficient solution of the problem. The decomposition principle, according to [16], also helps to tackle the truck and trailer problem.

The current chapter presents our results on vehicle navigation control. In Section 11.5, an overview of the complementing collision avoidance system proposed in [17] is given. Implementation issues and implications are discussed in Section 11.6 along with the results, including some brand new ones from combining collision avoidance system with truck and trailer control system.

## 11.2 Problem Formulation

Truck position is determined by three state variables  $x$ ,  $y$  and  $\Phi_c = [-90^\circ, 270^\circ]$ , where the latter is the angle between truck's onward direction and the  $x$ -axis (Figure 11.1). The width and length of the truck are denoted by  $w$  and  $l_c$ , respectively ( $w = 2$ ,  $l_c = 4$ ).

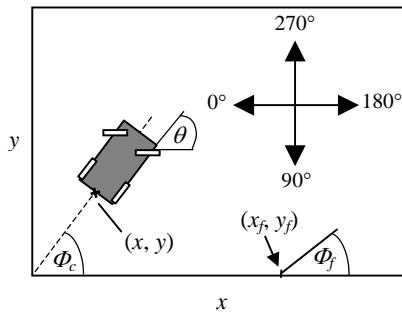


Figure 11.1. Truck and main variables

The problem is formulated as follows: the truck must travel from the arbitrary initial position  $(x_0, y_0, \Phi_0)$  to the predefined loading dock  $(x_f, y_f, \Phi_f)$ . To accomplish this, the appropriate steering angle  $\theta = [-45^\circ, 45^\circ] = f(x, y, \Phi_c)$  must be provided. Note that the truck moves forward or backward with fixed speed, and thus speed control is not our concern.

Truck kinematics are simulated using the following set of equations:

$$\begin{cases} \Phi_c(t + \Delta t) = \Phi_c(t) + (v \cdot \Delta t / l) \tan \theta \\ x(t + \Delta t) = x(t) + v \cdot \Delta t \cos(\Phi_c(t)) \\ y(t + \Delta t) = y(t) + v \cdot \Delta t \sin(\Phi_c(t)) \end{cases}, \quad (11.1)$$

where  $v$  is the speed of the truck and  $\Delta t$  is the sampling time.

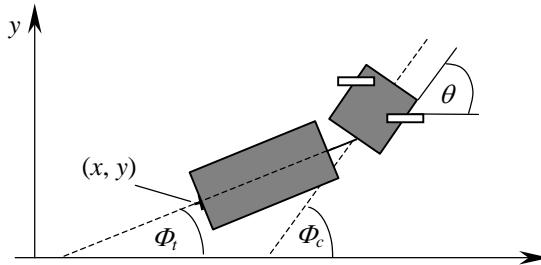


Figure 11.2. Truck and trailer

In the full truck backer-upper problem (Figure 11.2), a trailer is connected to the truck. Inclusion of the trailer brings an additional state variable, the angle of the trailer ( $\Phi_t$ ). The coordinate point of the moving object is transferred from the cab part to the backside of the trailer because the control goal now is to guarantee proper positioning of the trailer part. It is also assumed that  $|\Phi_t - \Phi_c| < 60^\circ$  and in this system,  $\theta = [-70^\circ, 70^\circ]$ . Length and width of the trailer are  $l_t = 4$  and  $w = 2$  meters, respectively. The dimensions ( $w \times l_c$ ) of the cab part are  $2 \times 2$  meters<sup>2</sup>. Other assumptions and control objectives remain unchanged. Current implementation uses the set of equations from [3]

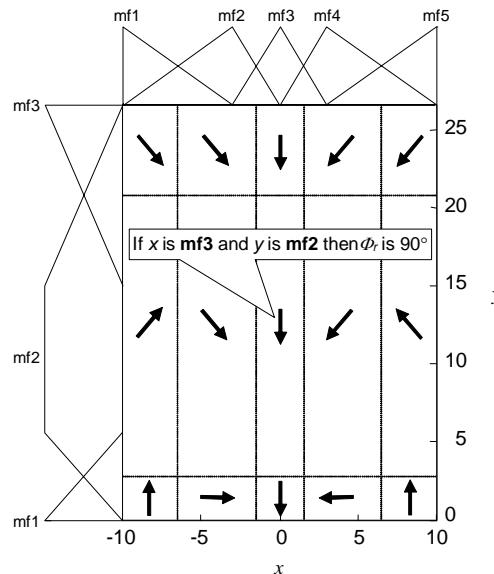
$$\begin{cases} x(t + \Delta t) = x(t) - B \cos(\Phi_t(t)) \\ y(t + \Delta t) = y(t) - B \sin(\Phi_t(t)) \\ \Phi_t(t + \Delta t) = \Phi_t(t) - \arcsin((A \sin(\Phi_c(t) - \Phi_t(t)) / l_t)) \\ \Phi_c(t + \Delta t) = \Phi_c(t) - \arcsin((v \cdot \Delta t / (l_t + l_c)) \sin(\theta)) \end{cases}, \quad (11.2)$$

where  $A = v \cdot \Delta t \cos(\theta)$ ,  $B = A \cos(\Phi_c(t) - \Phi_t(t))$ .

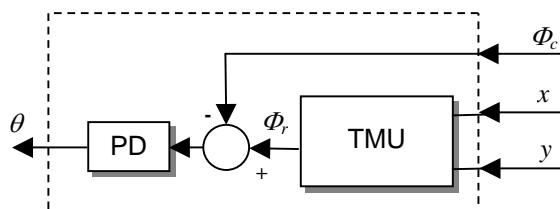
### 11.3 Vehicle Navigation Control

The most important component of the control system is the trajectory mapping unit (tmu) that provides optimal truck angle ( $\Phi_t$ ) for the given point in input space determined by its current coordinates  $x$  and  $y$ , to reach the loading dock situated at ( $x_f = 0$ ,  $y_f = 0$ ,  $\Phi_f = 90^\circ$ ). The information encapsulated in the tmu is very intuitive and can therefore be easily specified as can be seen from Figure 11.3.

Computation of the actual steering angle  $\theta$  that would ensure desired car orientation  $\Phi_r$  is carried out by a PD control loop (Figure 11.4). The advantages of such configuration when compared to all-in-one approaches [7,12], where fuzzy controller is required to perform the function  $\theta = f(x, y, \Phi_r)$  become obvious. First, the number of input variables to the fuzzy system is reduced (which allows us to use  $y$  (usually neglected for the sake of simplicity in such contributions). Second, the structure of the control system facilitates more efficient control knowledge acquisition and implementation. Finally, feedback in the control loop stabilizes and enhances steering quality.



**Figure 11.3.** Fifteen tmu rules responsible for trajectory management. Each small arrow indicates optimal angle of the truck corresponding to the given rule



**Figure 11.4.** Hierarchical control system

To expand functionality of the control system, the following modifications are made. First, in order to expand its rather small working range ( $x = [-10, 10]$ ,  $y = [0, 27]$ ), the saturation functions are used for input variables,  $x$  and  $y$ , which limit the input values to the upper and lower bounds of the tmu variables to ensure that car

navigation will be governed by appropriate rules even when  $x$  and  $y$  appear to be outside the scope of the tmu.

Second, if  $(x_f \neq 0, y_f \neq 0, \Phi_f \neq 90^\circ)$  then in order to save ourselves from retuning the tmu each time the final destination is changed, the original tmu in Figure 11.4 is replaced by the one depicted in Figure 11.5 that comes with built-in interface between the actual values of variables  $x$ ,  $y$ ,  $\Phi_c$  and those that will be used for the computations in the interfaced tmu (which are denoted by  $x'$ ,  $y'$  and  $\Phi'_c$ ).

The values of  $x'$  and  $y'$  are obtained using the following formulae:

$$\begin{aligned} x' &= r \cos(90^\circ - \Phi_f + \Phi_c) \\ y' &= r \sin(90^\circ - \Phi_f + \Phi_c) \end{aligned} \quad (11.3)$$

where

$$r = \sqrt{(x - x_f)^2 + (y - y_f)^2} . \quad (11.4)$$

Setpoint value corresponding to the given destination  $(x_f, y_f, \Phi_f)$  is computed by

$$\Phi_r = \Phi'_r + \Phi_f - 90^\circ . \quad (11.5)$$

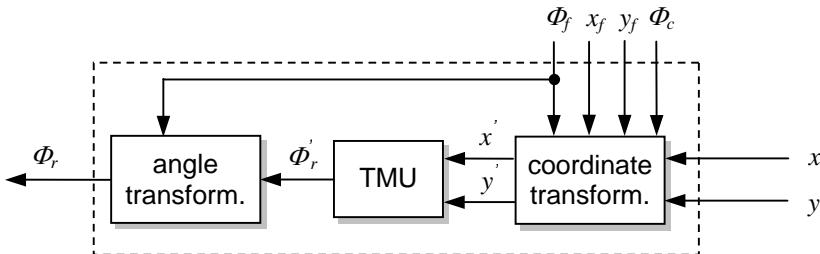
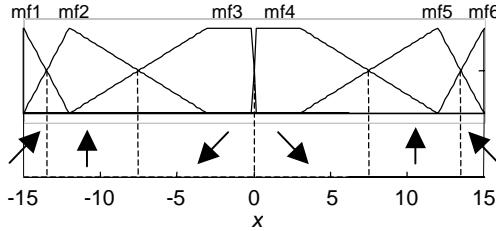


Figure 11.5. Generalized tmu interface

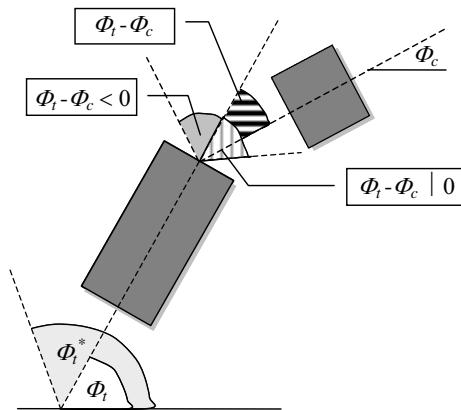
Third, the tmu should also be extended to cover the negative range of values of  $y$  where different actions have to be taken to bring the truck home. To avoid undesirable co-effects from the interpolation of antagonistic rules, the part of the tmu that corresponds to the range of negative values of  $y$ , is realized as a separate piece (Figure 11.6).

Note that the expanded system can also be used for truck control in forward driving mode if we correct  $\Phi_r$  by  $180^\circ$  (the reason is obvious) and multiply the PD controller output  $\theta$  by  $-1$  (the latter is necessary because turning the wheels to the left would turn the car to the right in backward driving mode but would have the opposite result in forward driving mode).



**Figure 11.6.** Tmu rules responsible for trajectory management when  $y < 0$

Also note that the control system is currently optimized to truck's physical parameters specified in Section 11.2. Although the control system is quite robust, substantial increase of car dimensions and/or smaller maximum steering angle would both hamper car maneuverability, which may result in a failure. If this is the case, it is possible to adjust the tmu to the changes with the help of scaling factors applied to tmu inputs  $x$  and  $y$ . In fact, these scaling factors already come in handy in the next section, in the truck and trailer control system.

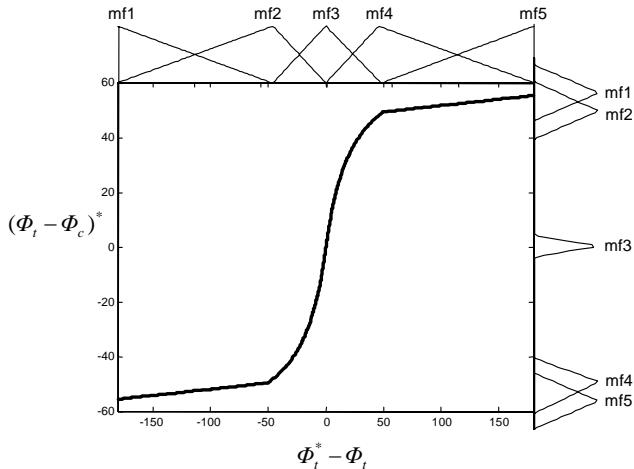


**Figure 11.7.** Relationship between truck and trailer angles

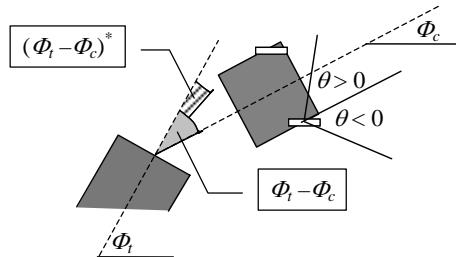
## 11.4 Truck and Trailer Backing Control

With truck and trailer, the output of the tmu is interpreted as desired trailer angle  $\Phi_t^*$ . From Figure 11.7 it is obvious that if trailer orientation error  $\Phi_t^* - \Phi_t$  is positive, we must increase  $\Phi_t$  (and similarly decrease it if the error is negative) in order to reduce the error.  $\Phi_t$  grows when  $\Phi_t - \Phi_c$  remains positive. To establish this, a simple joint controller that computes appropriate  $(\Phi_t - \Phi_c)^*$ , desired difference between truck and trailer angles against trailer orientation angle, is specified (Figure 11.8). According to our reasoning, the function realized by the joint controller must be growing monotonously. We have implemented the highly nonlinear curve by the means of fuzzy logic where final membership function

values were found by manual tuning during the backing (as so few variables and rules (five) are involved, the procedure is not overly complicated).



**Figure 11.8.** The joint controller



**Figure 11.9.** Angles around the joint

Steering angle can again be computed by PD control loop; however, we need to determine the its sign first – *i.e.* if the actual difference between the truck and the trailer  $\Phi_t - \Phi_c$  is larger than the desired one  $(\Phi_t - \Phi_c)^*$ , we must increase  $\Phi_c$  and thus  $\theta$  must be negative in this case (Figure 11.9).

The resulting control system is depicted in Figure 11.10. Note that the parameters of the PD controller are  $K_p = -12$ ,  $K_d = -4$  (because of the sign issue) and that variables  $x$  and  $y$  have been scaled by  $k_x = k_y = 0.8$ .

## 11.5 Collision Avoidance System

As such, the vehicle, or more appropriately, its control system, is ignorant of possible obstacles coming its way (including the bounding walls) and is therefore

unable to avoid consequent collisions. This section explains how the problem can be solved through some extension of the original control system.

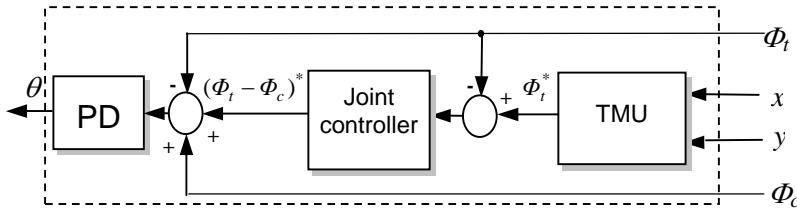


Figure 11.10. Truck and trailer backer-upper control system

For obstacle avoidance the truck must first be equipped with the devices that make possible object detection. Eight distance sensors are “installed” to the truck body and in case of a truck and trailer system – to the trailer body (although only four of them will be active at a time): left and right rear sensors  $s_1$  and  $s_2$ , right and left side sensors  $s_3$  -  $s_6$  and left and ride frontal sensors  $s_7$  and  $s_8$  giving readings  $d_1$  -  $d_8$ , respectively (Figure 11.11).

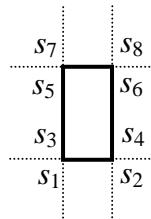


Figure 11.11. Distance sensor placement scheme

Collision avoidance in backward driving mode works on the following principle: if an obstacle is detected in the truck’s way (either  $d_1$  or  $d_2$  is smaller than a predetermined detection limit  $d_{\text{det}}$ ), first, the turning direction (left or right) is determined on the basis of  $d_1$  -  $d_4$  (Figure 11.12).

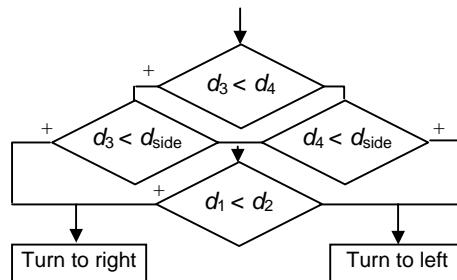


Figure 11.12. Turning decision making

Based on this decision and orientation of the obstacle (that will be identified according to rear distance sensor readings) a temporary gate is projected onto the bounding wall in the car's turning direction (Figure 11.13). Note that presently, turning direction is chosen according to rear sensor readings only, as there are no relevant obstacles in the car side directions. Also note that the opposite turning decision would result in the temporary gate projected onto opposite (upper) bounding wall.

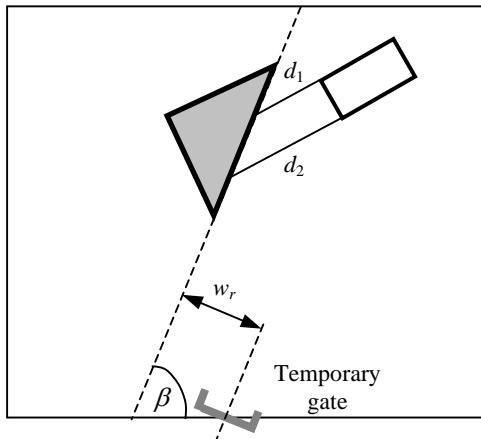


Figure 11.13. Temporary driving goal

In forward driving mode the algorithm is basically the same, only we are using sensors  $s_5 - s_8$  instead of  $s_4, s_3, s_2$  and  $s_1$ , respectively and the obstacle orientation angle  $\beta$  must be corrected by  $180^\circ$ .

Performance of the collision avoidance system is subject to the following parameters:

$d_{\text{det}}$  – Obstacle detection limit

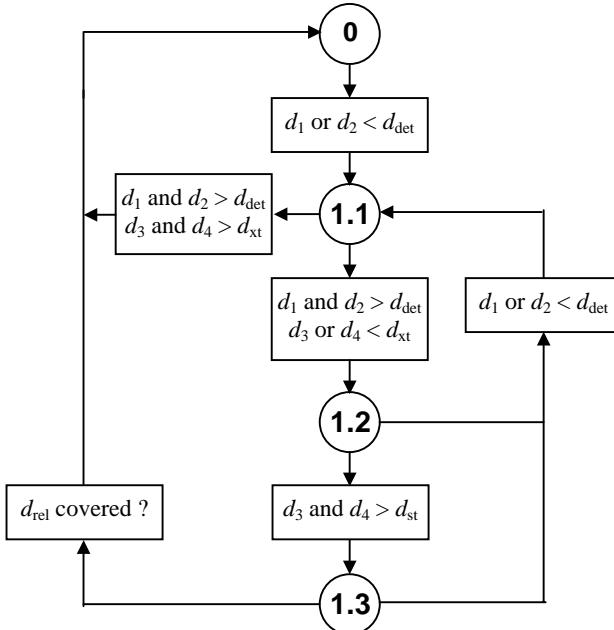
$d_{\text{excl}}$  – Rear sensor readings larger than  $d_{\text{excl}}$  will be emulated by  $d + w$ , where  $d$  is the rear sensor reading responsible for obstacle detection

$d_{\text{side}}$  – Determines if side sensor information will be used in turning direction determination (Figure 11.12)

$w_r$  – The distance by which the temporary gate is shifted from the obstacle-projected line (Figure 11.13)

### 11.5.1 State Transition Control

We can speak of two states in car control – normally the car moves toward  $(x_f, y_f, \Phi_f)$  according to the tmu-specified profile (state 0); however, when an obstacle is detected by distance sensors, the collision avoidance system (previous section) takes over and switches the tmu to the temporary gate (state 1). This temporary goal remains valid until it is safe to return to the normal state. For overall good performance, state transition control must be smooth and effective. This is achieved in three steps as explained below.



**Figure 11.14.** State transition diagram (backward driving mode)

The collision avoidance system takes over if at least one of the sensors in the moving direction of the car registers a distance smaller than  $d_{\text{det}}$  (transition from state 0 to 1.1 in Figures 11.14 and 11.15). The car then starts to rotate because of the temporary driving goal that has been imposed in the moment of state transition and further state management will be delegated to side sensors if the obstacle comes within reach of one of them (substate 1.2), unless a new obstacle is detected and a new temporary gate must be computed (return to substate 1.1). Eventual return to state 0 can principally happen if the obstacle is no longer present at the car's side (substate 1.3). In certain circumstances, however, it may be necessary to delay the release further (too early release may cause a collision with the obstacle currently being passed), which is controlled by  $d_{\text{rel}}$  – small distance that has to be covered while in substate 1.3.

State transition is independently regulated by two extra parameters ( $d_{\text{min}}$  and  $d_{\text{max}}$ ) that determine minimum and maximum time that the system may spend in state 1, respectively. The aim of  $d_{\text{min}}$  is to make car control less sensitive to rapid changes of external conditions. The need for  $d_{\text{max}}$  becomes apparent when a bounding wall or connected obstacles happen to be in the car's way because in this case the car cannot break out from state 1 on its own, even if it would be good for the overall goal.

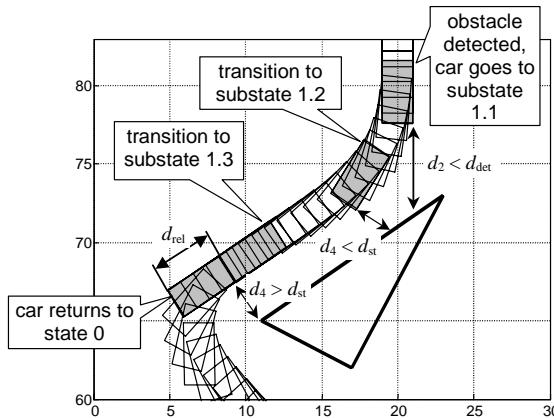


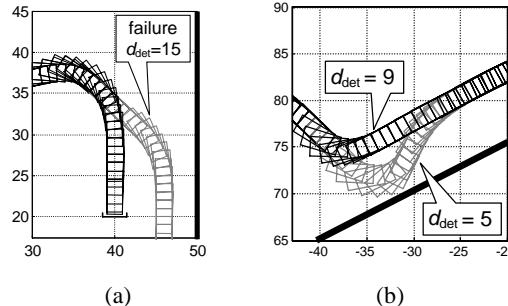
Figure 11.15. State transition in obstacle avoidance

## 11.6 Results and Discussion

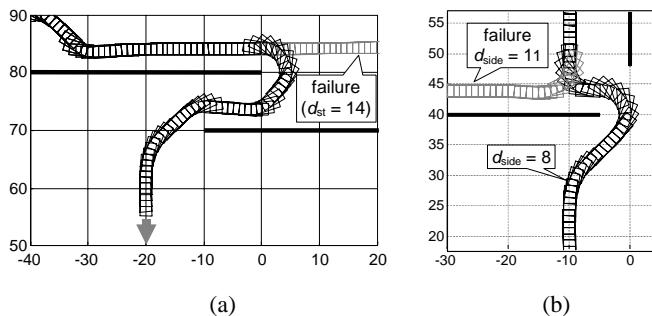
In this section simulated results with the designed navigation and collision avoidance system (mostly in the more challenging backward driving mode) obtained in various environments are presented. One of these environments contains obstacles of various shapes (Figure 11.18), the other two are the labyrinths (Figures 11.19a and 11.20). In order to perform the requested tasks we need to determine proper settings of  $d_{\text{side}}$ ,  $d_{\text{det}}$ ,  $d_{\text{excl}}$ ,  $d_{\text{st}}$ ,  $w_r$ ,  $d_{\text{rel}}$ ,  $d_{\text{min}}$  and  $d_{\text{max}}$ . These values mostly depend on the dimensions of the controlled object and the nature of the task (character of obstacles). Note that the listed parameters influence only the performance of the collision avoidance system, as the ability to reach the preset destination depends solely on the tmu which has currently been optimized for given truck or truck and trailer system dimensions.

It is possible to detect internal dependencies in parameter settings.  $w_r$  must be chosen such that the car would leave a reasonable distance between itself and the obstacle wall (about a few car widths), presently it is 4 m for the truck and 6 m for the truck and trailer system. The value of the detection limit  $d_{\text{det}}$  partially derives from the former *i.e.* it must be large enough (in current setting this parameter equals 6 and 12 m, respectively) to provide a smooth trajectory from the detection point to the point where  $w_r$  is established (Figure 11.16a). On the other hand, it is important to keep this parameter as small as possible so as not to make the car to react to the obstacles at longer distances (Figure 11.16b). Obviously,  $d_{\text{st}}$  (in current setting 9 and 15 m) must be larger than  $w_r$  to make sense and similar to  $d_{\text{det}}$  (the reason becomes apparent in Figure 11.17a).  $d_{\text{side}}$  must also be in the same range (in current setting 8 and 11 m) as a too large value can lead to incorrect strategical decisions (see Figure 11.17b). Release distance  $d_{\text{rel}}$  is equal to 0 in backward driving mode but 2 m in forward driving mode because the turning radius of the car is smaller in the latter case. The remaining parameters,  $d_{\text{min}} = 5$ ,  $d_{\text{max}} = 80$ ,  $d_{\text{excl}} = 30$ , are obtained rather empirically and/or in a few test drives.

Numerically, the driving results depicted in Figures 11.18 ~ 11.20 can be evaluated by the combined measure proposed in [15]. This measure, however, evaluates the final position of the car that primarily depends on the tmu (its ability to provide smooth and effective car trajectories has been documented in [15, 16]). The performance of the collision avoidance system has binary character (success/failure) and can easily be identified from respective Figures.



**Figure 11.16.** Settings for  $d_{det}$  and  $d_{side}$ : (a) too large value of  $d_{det}$  may connect the car with irrelevant obstacles (wall at right); (b) non-smooth trajectory in substate 1.1 resulting from too small a value of  $d_{det}$  (compared to  $w_r = 6$ )



**Figure 11.17.** Settings for  $d_{st}$  and  $d_{side}$ : (a) car does not come out of substate 1.2 if  $d_{st}$  is too large; (b) car fails to break out from the maze if the setting of  $d_{side}$  is improper

Despite good performance in presented simulations, it is clear that there exist limits on what the proposed control system can do. First, the test assignments are based on the reasonable assumption that they must be solvable under current capacities of the control system; *e.g.* there must be enough space for the car to perform necessary maneuvers.

Moreover, although the designed subsystems have highly cooperative natures, there are situations where it is impossible to perform the expected task because of the conflict of interest of theirs. Figure 11.19b presents one such case. At point *a*, the obstacle is recognized and the car starts its usual routine to avoid it. At point *b*, it encounters another obstacle and acts accordingly. The car is released at point *c* but because of the position of the global goal (loading dock) the car performs a left

turn and returns to the point it has already visited, apparently unable to find a solution to the problem.

For such situations, another extension or “higher level of intelligence” of the control system is needed. It should be able to recognize a failure and make more general decisions about the actions to be taken, *i.e.* when the car recognizes that it arrives in the position it has been before, it should force the system to solve the problem in alternative way and apply it in the right moment (in point *c*).

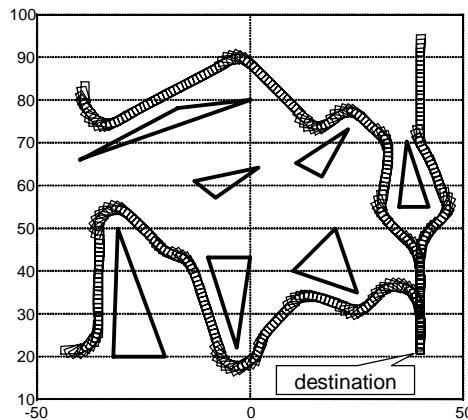


Figure 11.18. Driving the truck backwards in the field

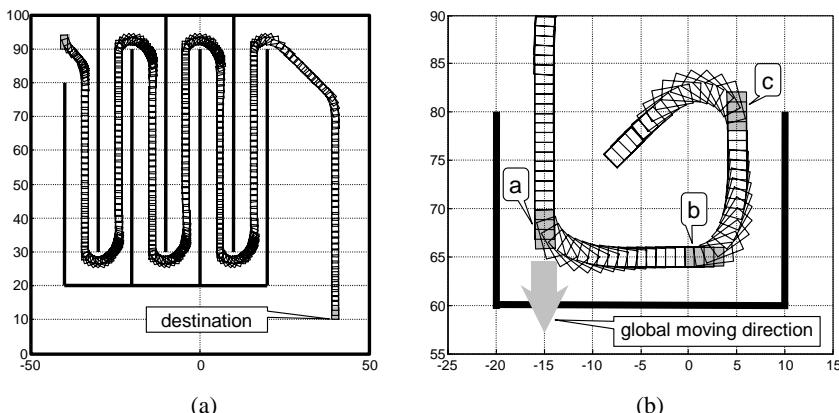
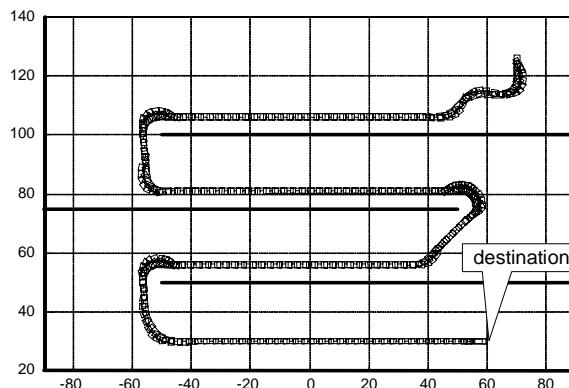


Figure 11.19. Results: (a) forward-driven truck in the labyrinth; (b) a possible failure situation

Second, the existing system performs in the noise-free environment. In real life, however, all computations that depend on measured data are subject to noise. This affects both car position determination and decisions based on distance sensor information.

The material in this chapter appears as the groundwork for car (or similar moving objects, such as mobile robots) navigation system model-scale

implementation and although it is possible to simulate the noise as well, at this stage of the project we do not know yet what kind of noise corresponds to the conditions we are going to encounter in real life. It is possible to make the existing system less sensitive to noise if the need arises, either by sensor information processing, appropriate tmu modification or more conservative (*i.e.* larger) system parameter values (this actually has been tested with white noise and found to be useful for collision avoidance system).



**Figure 11.20.** Backward-driven truck and trailer system performance in the labyrinth

## 11.7 Conclusions

In this chapter, a fuzzy logic enhanced navigation and collision avoidance system for truck or truck and trailer systems has been described. It assumes that vehicle coordinates and its orientation in respect to the  $x$ -axis can be determined and is based on the fuzzy trajectory mapping unit that provides smooth car trajectory management independent of the object's initial position or the position of the loading dock. It is suggested that for performing more demanding tasks than demonstrated here, additional blocks adding more "intelligence" are required. The latter is made possible by the modular structure of the control system that also enables us to tune subsystems of the control system responsible for different tasks individually without jeopardizing overall performance.

## Acknowledgement

The research has been partially supported by Estonian Science Foundation, grant no. 5170.

## References

- [1] D. Nguyen and B. Widrow, "The truck backer-upper: An example of self-learning in neural network," *IEEE Contr. Syst. Mag.*, 1990, Vol. 10, No. 2, pp 18-23.
- [2] J.R. Koza, "A genetic approach to the truck backer upper problem and the intertwined spirals problem". *Proc. Int. Joint Conf. Neural Networks*, Piscataway, NJ, Vol. 4, 1992, pp 310-318.
- [3] M. Schoenauer and E. Ronald, "Neuro-genetic truck backer-upper controller. Proc. *First Int. Conf. Evolutionary Comp.*, Orlando, FL, USA, 1999, pp 720-723.
- [4] R.E. Jenkins and B.P. Yuhas, "A Simplified Neural Network Solution Through Problem Decomposition: The Case of the Truck Backer-Upper", *IEEE Trans. Neural Networks*, Vol. 4, No. 4, 1993, pp 718-720.
- [5] S. Kong and B. Kosko, "Comparison of fuzzy and neural truck backer-upper control systems". *Proc. IJCNN*, Vol. 3, 1990, pp 349-358.
- [6] K. Tanaka, T. Kosaki and H.O. Wang, "Backing Control Problem of a Mobile Robot with Multiple Trailers: Fuzzy Modeling and LMI-Based Design". *IEEE Trans. Syst., Man, Cybern. Part C*, Vol. 28, No. 3, 1998, pp 329-337.
- [7] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. on System, Man, and Cybernetics*, Vol. 22, No. 6, 1992, pp 1414-1427.
- [8] P.A. Ramamoorthy and S. Huang, "Fuzzy Expert Systems vs. Neural Networks – Truck Backer-Upper Control Revisited," *Proc. IEEE*, 1991, pp 221-224.
- [9] J.J. Shann and H.C. Fu, "A fuzzy neural network for rule acquiring on fuzzy control systems", *Fuzzy Sets and Systems*, Vol. 71, 1995, pp 345-357.
- [10] A. Ismail and E.A.G. Abu-Khousa, "A Comparative Study of Fuzzy Logic and Neural Network Control of the Truck Backer-Upper System," *Proc. IEEE Int. Symp. on Intelligent Control*, Dearborn, 1996, pp 520-523.
- [11] D. Kim, "Improving the fuzzy system performance by fuzzy system ensemble", *Fuzzy Sets and Systems*, Vol. 98, 1998, pp 43-56.
- [12] S.-G. Kong, B. Kosko, "Adaptive Fuzzy Systems for Backing up a Truck-and-Trailer," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, 1992, pp 211-223.
- [13] I. Dumitache and B. Catalin, "Genetic learning of fuzzy controllers," *Mathematics and Computers in Simulation*, Vol. 49, 1999, pp 13-26.
- [14] M.-C. Su and H.-T. Chang, "Application of neural networks incorporated with real-valued genetic algorithms in knowledge acquisition," *Fuzzy Sets and Systems*, Vol. 112, 2000, pp 85-97.
- [15] A. Riid and E. Rüstern, "Fuzzy logic in control: truck backer-upper problem revisited" *Proc. 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Vol. 1, 2001, pp 513-516.
- [16] A. Riid and E. Rüstern, "Fuzzy Hierarchical Control of Truck and Trailer" *Proc. 8th Biennal Baltic Electronic Conf.*, Tallinn, Estonia, 2002, pp 141-144.
- [17] A. Riid and E. Rüstern, "Car Navigation and Collision Avoidance System with Fuzzy Logic", *Proc IEEE International Conference on Fuzzy Systems*, Budapest, Vol. 3, 2004, pp 1443-1448.

## **Fuzzy Logic Based Control Mechanisms for Handling the Uncertainties Facing Mobile Robots in Changing Unstructured Environments**

Hani Hagras

### **12.1 Introduction**

Autonomous mobile robots navigating in real-world changing and dynamic unstructured environments (like outdoor environments) need adequate control mechanisms to handle the uncertainties and imprecision present in such environments.

The fuzzy logic controller (FLC) is credited with being an adequate methodology for designing robust controllers that are able to deliver a satisfactory performance in face of uncertainty and imprecision [10]. The use of fuzzy rules and linguistic variables makes fuzzy control an adequate design tool for non-linear systems for which a precise mathematical model cannot be easily obtained, but for which heuristic control knowledge is available which is the case of mobile robots navigating in unstructured environments. As a result the FLC has become a popular approach to mobile robot control in recent years [10-12].

There are many sources of uncertainty facing the FLC for a mobile robot navigating in changing and dynamic unstructured environments; we list some of them as follows:

- Uncertainties in inputs to the FLC which translates into uncertainties in the antecedents membership functions as the sensors measurements are typically noisy and are affected by the conditions of observation (*i.e.* their characteristics are changed by the environmental conditions such as wind, sunshine, humidity, rain, *etc.*). These input uncertainties cause difficulty in determining the antecedents fuzzy membership functions, as if they are determined and tuned in certain environmental conditions, they might have to be changed in other environments.
- Uncertainties in control actions which translates into uncertainties in the output membership functions of the FLC. Such uncertainties can result from the change of the actuators characteristics which can be due to the inconsistency of the terrain or due to the environmental changes. For example, a “fast speed” on a sunny day with dry ground may be different to “fast speed” on a rainy day with muddy ground as wheels may slip *etc.*

Another source of uncertainty in the control actions is caused by changes of the mobile robot's physical properties such as changing of the diameter of the wheels, belts loosening due to wear and tear, etc. These output uncertainties cause difficulty in determining the consequents fuzzy membership functions; as if determined and tuned in certain environmental and robot conditions, they might have to be changed in other circumstances.

- Linguistic uncertainties as to the meaning of words that are used in the antecedents and consequents linguistic labels can be a problem as words mean different things to different people [8]. In addition, experts don't always agree and they often provide different consequents for the same antecedents. A survey of experts will usually lead to a histogram of possibilities for the consequent of a rule, this histogram representing the uncertainty about the consequent of a rule [7].

All of these uncertainties translate into uncertainties about fuzzy set membership functions [8].

To date all the FLC implementations in robot control are based on the traditional type-1 FLC. Type-1 FLC uses type-1 fuzzy sets which handles the uncertainties associated with the inputs and outputs by using *precise and crisp* membership functions that the user believes capture the uncertainties. Once the type-1 membership functions have been chosen, all the uncertainty disappears, because type-1 membership functions are totally precise [7]. According to Klir and Fogler [3], "it may seem problematical, if not paradoxical, that a representation of fuzziness is made using membership grades that are themselves precise real numbers".

The type-1 FLCs have the common problem that they cannot model or accommodate the uncertainties associated with changing and dynamic unstructured environments as they use crisp type-1 fuzzy sets. These uncertainties cause problems in determining the exact antecedents and consequent membership functions and rules and we might end up wasting time in redesigning or tuning the FLC so that it can deal with the various uncertainties faced in changing unstructured environments.

A type-2 fuzzy set is characterized by a fuzzy membership function, *i.e.* the membership value (or membership grade) for each element of this set is a fuzzy set in  $[0,1]$ , unlike a type-1 fuzzy set where the membership grade is a crisp number in  $[0,1]$  [7]. The membership functions of type-2 fuzzy sets are three-dimensional, it is the new third-dimension of type-2 fuzzy sets that provides additional degrees of freedom that make it possible to model uncertainties directly [8]. The type-2 fuzzy sets are useful where it is difficult to determine the exact membership functions (which is the case for mobile robots in unstructured environments); hence, they are useful for incorporating and handling uncertainties [4]. Although a type-2 membership function will also be totally precise, it includes the footprint of uncertainty (FOU) that provides new degrees of freedom that let uncertainties be handled in totally new ways [7]. The type-2 FLC presented in this chapter uses type-2 fuzzy sets for the input and output membership functions. The type-2 FLC produces a reduction in the rule base compared to the type-1 FLC as the footprint

of uncertainty provided by the type-2 fuzzy sets allows us to cover the same range with a much smaller number of labels [7]. In addition, the type-2 FLC will be able to model and handle the uncertainties faced by the mobile robots in changing and dynamic unstructured environments and hence it will have the potential to produce a better performance than the type-1 FLC as will be demonstrated through our experiments in indoor and outdoor changing and dynamic unstructured environments.

To the author's knowledge, no other work in the literature had investigated the use of type-2 FLC to mobile robot real time control in unstructured environments. There is only one relevant paper produced by Wu [13] which used fuzzy interval control but they used type-1 fuzzy sets and type-1 FLC where the type-1 fuzzy sets were adaptive as they used a set of parameters called sensitivity indices which determine the support and domain of the membership functions. However, they did not address the use of type-2 fuzzy sets and how to implement a real time type-2 FLC for a mobile robot navigating in unstructured dynamic environments like outdoor environments.

This chapter is organised as follows. In Section 12.2, we will review the type-2 fuzzy sets, interval type-2 fuzzy sets and their associated terminologies as well as the type-2 set theoretic operations. In Section 12.3, we will introduce the interval type-2 FLC and its various components. In Section 12.4, we will present the experiments and results using real robots navigating in indoor and outdoor unstructured environments. Finally, conclusions will be presented in Section 12.5.

## 12.2 Type-2 Fuzzy Sets

Type-2 fuzzy sets are able to model the linguistic and numerical uncertainties faced by the robot FLC because their membership functions are themselves fuzzy [8]. According to [8] imagine blurring the type-1 membership function depicted in Figure 12.1a by shifting the points on the triangle either to the left or to the right and not necessarily by the same amounts, as in Figure 12.1b. Then, at a specific value of  $x$ , say  $x'$ , there is no longer a single value for the membership function ( $u'$ ); instead, the membership function takes on values wherever the vertical line intersects the blurred area shaded in grey. Those values need not all be weighted the same; hence, 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, which characterizes a type-2 fuzzy set.

A type-2 fuzzy set, denoted  $\tilde{A}$  is characterised by a type-2 membership function  $\mu_{\tilde{A}}(x, u)$  [8], where  $x \in X$  and  $u \in J_x$  in which  $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$ .  $\tilde{A}$  can be expressed as follows [8]:

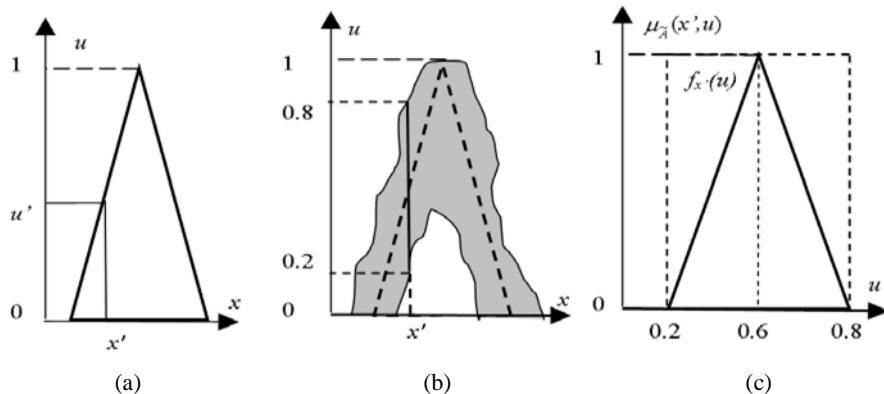
$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0, 1] \quad (12.1)$$

where  $\int \int$  denotes union over all admissible  $x$  and  $u$  [8].

At each value of  $x$ , say  $x = x'$ , the 2-D plane whose axes are  $u$  and  $\mu_{\tilde{A}}(x', u)$  is called a vertical slice of  $\mu_{\tilde{A}}(x, u)$  [8]. A *secondary membership function* is a vertical slice of  $\mu_{\tilde{A}}(x, u)$ . It is  $\mu_{\tilde{A}}(x=x', u)$  for  $x' \in X$  and  $\forall u \in J_{x'} \subseteq [0, 1]$ :

$$\mu_{\tilde{A}}(x=x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/(u) \quad J_{x'} \subseteq [0, 1] \quad (12.2)$$

in which  $0 \leq f_{x'}(u) \leq 1$  [8]. Because  $\forall x' \in X$ , the prime notation on  $\mu_{\tilde{A}}(x')$  is dropped and  $\mu_{\tilde{A}}(x)$  is referred to as a secondary membership function [8]; it is a type-1 fuzzy set which is also referred to as a *secondary set* [8]. Many choices are possible for the secondary membership functions. According to Mendel [7] the name that we use to describe the entire type-2 membership function is associated with the name of the secondary membership functions; so, for example if  $f_x(u)=1$ ,  $\forall u \in J_x \subseteq [0, 1]$ , then the secondary membership functions are interval sets (as shown in Figure 12.1c), and, if this is true for  $\forall x \in X$ , we have the case of an *interval type-2 membership function* which characterises the interval type-2 fuzzy sets [7]. According to Liang *et al.* [4] a type-2 fuzzy set  $\tilde{A}$  can be thought of as a large collection of embedded type-1 sets, each having a weight associated with it.



**Figure 12.1.** (a) Type-1 membership function. (b) Blurred type-1 membership function. (c) Triangular secondary membership function plotted in *thick line*; interval secondary membership function plotted in *dashed line*

Using type-2 fuzzy sets to represent the inputs and outputs of a mobile robot, FLC has many advantages when compared to the type-1 fuzzy sets; we summarize some of these advantages as follows:

- As the type-2 fuzzy sets membership functions are themselves fuzzy and include a FOU, then they can model and handle the linguistic and numerical uncertainties associated with the inputs and outputs of the robot FLC in

changing and dynamic unstructured environments and hence they can handle the difficulty associated with determining the exact membership functions for the fuzzy sets [4]. Therefore, FLCs that are based on type-2 fuzzy sets will have the potential to produce a better performance than the type-1 FLCs.

- Using type-2 fuzzy sets to represents the FLC inputs and outputs will result in the reduction of the FLC rule base when compared to using type-1 fuzzy sets as the uncertainty represented in the FOU in type-2 fuzzy sets lets us cover the same range as type-1 fuzzy sets with smaller number of labels and the rule reduction will be greater when the number of the FLC inputs increases [7].
- Each input and output will be represented by a large number of type-1 fuzzy sets which are embedded in the type-2 fuzzy sets. The use of such a large number of type-1 fuzzy sets to describe the input and output variables allows for a detailed description of the analytical control surface as the addition of the extra levels of classification gives a much smoother control surface and response [1].

So using type-2 fuzzy sets to represent the inputs and outputs of a mobile robot FLC enables us to model and handle uncertainties and can result in a better performance than when using type-1 fuzzy sets while using a lower number of rules; this will be justified through experiments using real robots navigating in challenging unstructured environments.

### 12.2.1 Interval Type-2 Fuzzy Sets

Interval type-2 sets are very useful when we have no other knowledge about secondary memberships [4]. The membership grades of the interval type-2 fuzzy sets are called “interval type-1 fuzzy sets” [4]. Since all the memberships in an interval type-1 set are unity, in the sequel, an interval type-1 set is represented just by its domain interval, which can be represented by its left and right end-points as  $[l, r]$  [4]. The two end-points are associated with two type-1 membership functions that are referred to as *upper* and *lower* membership functions [4].

The upper and lower membership functions are two type-1 membership functions which are bounds for the footprint of uncertainty  $FOU(\tilde{A})$  of a type-2 fuzzy set  $\tilde{A}$  [7]. The upper membership function is associated with the upper bound of  $FOU(\tilde{A})$  and is denoted by  $\bar{\mu}_{\tilde{A}}(x), \forall x \in X$  and can be written as follows [7]:

$$\bar{\mu}_{\tilde{A}}(x) \equiv \overline{FOU(\tilde{A})} \quad \forall x \in X \quad (12.3)$$

The lower membership function is associated with the lower bound of  $FOU(\tilde{A})$  and is denoted by  $\underline{\mu}_{\tilde{A}}(x), \forall x \in X$  and can be written as follows [8]:

$$\underline{\mu}_{\tilde{A}}(x) \equiv \underline{FOU(\tilde{A})} \quad \forall x \in X \quad (12.4)$$

According to Mendel [7] the interval type-2 fuzzy set  $\tilde{A}$  could be written in terms of upper and lower membership functions as follows:

$$\tilde{A} = \int_{x \in X} \left[ \int_{u \in [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]} \frac{1}{u} \right] / x \quad (12.5)$$

### 12.2.2 Type-2 Set Theoretic Operations

For type-2 fuzzy sets there are new operators named as the meet and the join to account for the intersection and union. Liang and Mendel [5] had derived the expressions for meet and join in interval type-2 fuzzy sets in which we need to compute the join, meet of secondary membership functions which are type-1 interval fuzzy sets [7].

$$\begin{aligned} \text{Let } F &= \int_{v \in F} 1/v \text{ be an interval type-1 set with domain } [l_f, r_f] \subseteq [0,1] \text{ and } G \\ &= \int_{w \in G} 1/w \text{ is another interval type-1 set with domain } [l_g, r_g] \subseteq [0,1]. \end{aligned}$$

The *meet*  $Q$  between  $F$  and  $G$  under the product t-norm which is used in our type-2 FLC is written as follows [5]:

$$Q = F \sqcap G = \int_{q \in [l_f l_g, r_f r_g]} 1/q \quad (12.6)$$

From Equation (12.6) each term in  $F \sqcap G$  is equal to the product of  $v.w$  for some  $v \in F$  and  $w \in G$  in which the smallest term is  $l_f l_g$  and the largest  $r_f r_g$  [7]. Since both  $F$  and  $G$  have continuous domains,  $F \sqcap G$  has a continuous domain, therefore  $F \sqcap G$  is an interval type-1 set with domain  $[l_f l_g, r_f r_g]$  [4]. In a similar manner the meet under product t-norm of  $n$  interval type-1 sets  $F_1, \dots, F_n$  having domains  $[l_1, r_1], \dots, [l_n, r_n]$  respectively, is an interval set with domain  $\left[ \prod_{o=1}^n l_o, \prod_{o=1}^n r_o \right]$ . The meet under minimum t-norm is calculated in a similar manner.

The join between  $F$  and  $G$  is given by

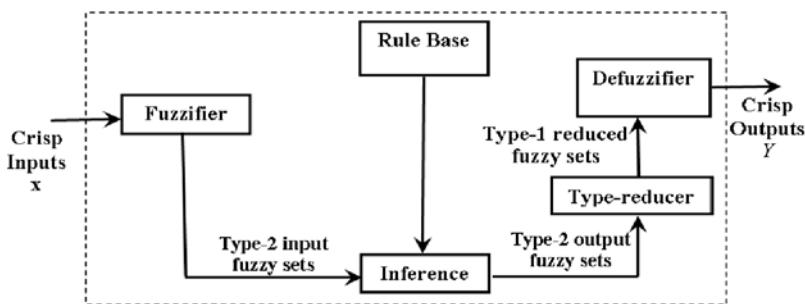
$$Q = F \sqcup G = \int_{q \in [l_f \vee l_g, r_f \vee r_g]} 1/q \quad (12.7)$$

where  $q = v \vee w$ , and  $\vee$  denotes the maximum operation used in out type-2 FLC. The join of  $n$  interval type-1 sets  $F_1, \dots, F_n$  having domains  $[l_1, r_1], \dots, [l_n, r_n]$  respectively is an interval set with domain  $[(l_1 \vee l_2 \vee \dots \vee l_n), (r_1 \vee r_2 \vee \dots \vee r_n)]$ , i.e. with domain equal  $[\max(l_1, l_2, \dots, l_n), \max(r_1, r_2, \dots, r_n)]$  [5].

### 12.3 Interval Type-2 Fuzzy Logic Controller (FLC)

An interval type-2 FLC is depicted in Figure 12.2; it contains five components which are fuzzifier, rule base, fuzzy inference engine, type-reducer and defuzzifier. In developing the type-2 FLC for mobile robot control both the inputs and outputs will be represented by type-2 fuzzy sets to handle the uncertainties associated with the input and output variables. We will use interval type-2 fuzzy sets to represent the input and output variables as they are simple to use [8] and they are very useful when we have no other knowledge about secondary memberships [4] (which is the case in robot control) as they distribute the uncertainty evenly among all admissible primary memberships [7]. Also, according to Mendel and John [8], at present it is very difficult to justify the use of other kinds of type-2 fuzzy sets, *e.g.*, as there is no best choice for type-1 fuzzy sets, so to compound this non-uniqueness by leaving the choice of the secondary membership functions arbitrarily is hardly justifiable [8]. Furthermore, the general type-2 FLC is computationally intensive [4] and the computation simplifies a lot when using interval type-2 FLC (using interval type-2 fuzzy sets).

The interval type-2 FLC works as follows, the crisp inputs from the input sensors are first fuzzified into, in general, input type-2 fuzzy sets (however, we will consider only singleton fuzzification) which then activate the inference engine and the rule base to produce output type-2 fuzzy sets. The type-2 fuzzy outputs of the inference engine are then processed by the type-reducer which combines the output sets and then performs a centroid calculation, which leads to type-1 fuzzy sets called type-reduced sets [7]. The defuzzifier can then defuzzify the type-reduced type-1 fuzzy outputs to produce crisp outputs to be fed to the actuators. Like the type-1 FLC, the type-2 FLC can be viewed as a mapping from crisp inputs to crisp outputs and can be expressed quantitatively as  $y = f(x)$ . However, a type-2 FLC has more design degrees of freedom than the type-1 FLC, because its type-2 fuzzy sets are described by more parameters than are type-1 fuzzy sets [7].



**Figure 12.2.** The interval type-2 fuzzy logic controller (FLC)

According to Karnik *et al.* [2] the type-reduced set of a type-2 FLC is the centroid of a type-2 fuzzy output set for the FLC; consequently, each element of the type-reduced set is the centroid of some type-1 set embedded in the output set of the type-2 FLC. Each of these embedded sets can be thought of as an output set

of some type-1 FLC and, correspondingly, the type-2 FLC can be thought of as a collection of many different type-1 FLCs. Each of these type-1 FLCs is *embedded* in the type-2 FLC, so the type-reduced set is a collection of the outputs of all the type-1 FLCs embedded in the type-2 FLC. In this way, the type-2 FLC has the potential to outperform the type-1 FLC as it is dealing with the uncertainty thorough different embedded type-1 FLCs. In the following subsection, we will introduce each block in the type-2 FLC

### 12.3.1 The Fuzzifier

The fuzzifier maps a crisp input vector with  $p$  inputs  $\mathbf{x} = (x_1, \dots, x_p)^T \in X_1 \times X_2 \dots \times X_p \equiv \mathbf{X}$  into input fuzzy sets; these fuzzy sets can, in general, be type-2 fuzzy input sets  $\tilde{A}_x$  [7]. However, we will use a singleton fuzzification as it is fast to compute and thus suitable for the robot real time operation. In the singleton fuzzification, the input fuzzy set has only a single point of nonzero membership [4], *i.e.* [7]:

$\tilde{A}_x$  is a type-2 fuzzy singleton if  $\mu_{\tilde{A}_x}(\mathbf{x}) = 1/1$  for  $\mathbf{x} = \mathbf{x}'$  and  $\mu_{\tilde{A}_x}(\mathbf{x}) = 1/0$  for all other  $\mathbf{x} \neq \mathbf{x}'$

### 12.3.2 Rule Base

According to Mendel [7], the rules will remain the same as in type-1 FLC but the antecedents and the consequents will be represented by type-2 fuzzy sets. Consider a robot type-2 FLC having  $p$  inputs  $x_1 \in X_1, \dots, x_p \in X_p$  and  $c$  outputs  $y_1 \in Y_1, \dots, Y_c \in Y_c$ . The rule base for this Multi Input Multi Output (*MIMO*) FLC with  $M$  rules can be written as follows:

$$R = \{R^1_{MIMO}, R^2_{MIMO}, \dots, R^M_{MIMO}\} \quad (12.8)$$

where the  $i$ th rule has the following format:

$$R^i_{MIMO} : \text{IF } x_1 \text{ is } \tilde{F}_1^i \dots \text{ and } x_p \text{ is } \tilde{F}_p^i \text{ THEN } y_1 \text{ is } \tilde{G}_1^i, \dots, y_c \text{ is } \tilde{G}_c^i \quad i=1, \dots, M \quad (12.9)$$

According to Mendel [7], as the type-2 FLC rule base will have the same structure as the type-1 FLC then we can use the results from [6] which states that a rule base of *Multi Inputs and Multi Outputs (MIMO)* can be considered as a group of *Multi Input Single Output (MISO)* rule bases

### 12.3.3 Fuzzy Inference Engine

The fuzzy inference engine block in Figure 12.2 gives a mapping from input type-2 sets to output type-2 sets [7]. In the inference engine, multiple antecedents in the rules are connected using the *Meet* operation, the membership grades in the input sets are combined with those in the output sets using the extended sup-star composition, and multiple rules are combined using the *Join* operation. Just as the sup-star composition is the backbone computation for type-1 FLC, the extended

sup-star composition is the backbone computation for a type-2 FLC [7]. The extended sup-star composition can be obtained simply by extending the type-1 sup-star composition by replacing type-1 membership functions by type-2 membership functions, the sup operation with *Join* operation and the t-norm operation with the *Meet* operation [7].

In the previous subsection we mentioned that a *MIMO* rule base can be considered as a group of *MISO* rule bases, so like [7] we will concentrate on the *MISO* rule base. Each rule in a *MISO* fuzzy rule base with  $M$  rules having  $p$  inputs  $x_1 \in X_1, \dots, x_p \in X_p$  and one output  $y_k \in Y_k$  can be written as follows [7]:

$$R_k^i \text{ MISO: IF } x_1 \text{ is } \tilde{F}_1^i \text{ and } \dots \text{ and } x_p \text{ is } \tilde{F}_p^i \text{ THEN } y_k \text{ is } \tilde{G}_k^i \quad i=1, \dots, M \quad (12.10)$$

$R_k^i \text{ MISO}$  is described by the membership function  $\mu_{R^i}(\mathbf{x}, y_k) = \mu_{R^i}(x_1, \dots, x_p, y_k)$  and according to [7]  $\mu_{R^i}(\mathbf{x}, y_k)$  can be written as follows:

$$\begin{aligned} \mu_{R^i}(\mathbf{x}, y_k) &= \mu_{\tilde{A}^i \rightarrow \tilde{G}_k^i}(\mathbf{x}, y_k) = \mu_{\tilde{F}_1^i}(x_1) \dots \mu_{\tilde{F}_p^i}(x_p) \mu_{\tilde{G}_k^i}(y_k) \\ &= [\mu_{\tilde{F}_1^i}(x_1) \dots \mu_{\tilde{F}_p^i}(x_p) \mu_{\tilde{G}_k^i}(y_k)] \end{aligned} \quad (12.11)$$

Each rule  $R_k^i \text{ MISO}$  determines a type-2 output fuzzy set  $\tilde{S}_k^i$  for the  $k$ th output  $y_k$ .  $\tilde{S}_k^i$  is obtained using the extended sup-star composition (denoted by  $\circ$ ) where  $\tilde{S}_k^i = \tilde{A}_x \circ R_k^i \text{ MISO}$ .

In our interval type-2 FLC we will use the meet under product t-norm, so according to [7] the result of the input and antecedent operations, which are contained in the firing set  $\mu_{\tilde{F}_a^i}(x_a) \equiv F^i(\mathbf{x}')$ , is an interval type-1 set, as follows:

$$F^i(\mathbf{x}') = [\underline{f}^i(\mathbf{x}'), \overline{f}^i(\mathbf{x}')] \equiv [\underline{f}^i, \overline{f}^i] \quad (12.12)$$

where

$$\underline{f}^i(\mathbf{x}') = \underline{\mu}_{\tilde{F}_1^i}(x_1') * \dots * \underline{\mu}_{\tilde{F}_p^i}(x_p') \quad (12.13)$$

and

$$\overline{f}^i(\mathbf{x}') = \overline{\mu}_{\tilde{F}_1^i}(x_1') * \dots * \overline{\mu}_{\tilde{F}_p^i}(x_p') \quad (12.14)$$

### 12.3.4 Type-reduction

Type-reduction was proposed in [2, 7]; it is called type-reduction because this operation takes us from the type-2 output sets of the inference engine to a type-1

set that is called the “type-reduced set”. These type-reduced sets are then defuzzified to obtain crisp outputs that are sent to the motors of the mobile robots.

As in [5] we will use the center of sets type reduction, as it has reasonable computational complexity that lies between computationally expensive centroid type-reduction and the simple height and modified height type-reductions which have problems when only one rule fires [7]. The type-reduced set  $Y_{\cos}$  using the center of sets type-reduction can be expressed as follows:

$$Y_{\cos}(\mathbf{x})_k = [y_{lk}, y_{rk}] = \int_{y_k^1 \in [y_{lk}^1, y_{rk}^1]} \dots \int_{y_k^M \in [y_{lk}^M, y_{rk}^M]} \int_{f^1 \in [\underline{f}^1, \bar{f}^1]} \dots \int_{f^M \in [\underline{f}^M, \bar{f}^M]} \frac{\sum_{i=1}^M f^i y_k^i}{\sum_{i=1}^M f^i} \quad (12.15)$$

where  $Y_{\cos}(\mathbf{x})_k$  for  $k$ th output is an interval set determined by its left most point  $y_{lk}$  and its right-most point  $y_{rk}$ .  $k = (1, \dots, c)$  and  $c$  is the total number of outputs for the FLC.  $y_k^i$  corresponds to the centroid of the type-2 interval consequent set  $\tilde{G}_k^i$  of the  $i$ th rule for the  $k$ th output,  $y_k^i$  is a type-1 interval fuzzy set determined by its left most point  $y_{lk}^i$  and its right most point  $y_{rk}^i$  [5].  $f^i$  denotes the firing strength (degree of firing) of the  $i$ th rule which is an interval type-1 set determined by its left most  $\underline{f}^i$  and right-most point  $\bar{f}^i$  [5] where  $\underline{f}^i$  is calculated using Equation (12.13) and  $\bar{f}^i$  is calculated using Equation (12.14).

The calculation of the type-reduced sets is divided into two stages; the first stage is the calculation of centroids of the type-2 interval consequent sets of each rule which is conducted ahead of time and before starting the robot FLC operation. For each output we determine the centroids of all the output type-2 interval fuzzy sets representing this output, then the centroid of the type-2 interval consequent set for the  $i$ th rule will be one of the pre-calculated centroids of type-2 output sets which corresponds to the rule consequent. To calculate the centroids of the output interval type-2 fuzzy sets we used the iterative procedure developed by [7].

The second stage of type-reduction happens each control cycle to calculate the type-reduced sets. For any output  $k$  to compute  $Y_{\cos}(\mathbf{x})_k$  in Equation (12.15) we need to compute its two end points  $y_{lk}$  and  $y_{rk}$  which can be found using the iterative procedure developed by [5, 7].

### 12.3.5 Defuzzification

The final crisp defuzzified output is found by finding the average of  $y_{lk}$  and  $y_{rk}$ , hence the defuzzified crisp output for each output  $k$  is [5]

$$Y_k(\mathbf{x}) = \frac{y_{lk} + y_{rk}}{2} \quad (12.16)$$

The iterative procedure for the type-reduction is proven [5, 14] to converge in no more than  $M$  iterations to find  $y_{rk}$  and no more than  $M$  iterations to find  $y_{lk}$  where  $M$  is the number of rules. As the rule base of the robot controller gets larger, it will take a long time for each control cycle to converge to the type-reduced set, thus limiting the real time application of the type-2 FLC [14]. To cope with this problem we developed a hierarchical architecture [9] in which we break down the input space for analysis by sharing it amongst multiple low level type-2 behaviours.

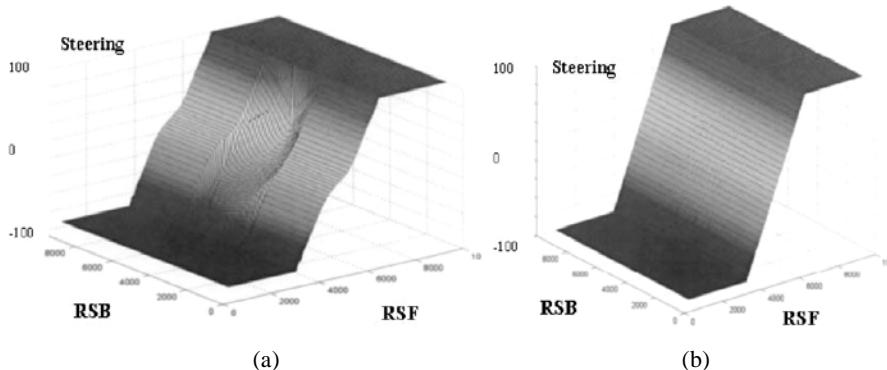
## 12.4 Experiments and Results

We will start by showing how the type-2 FLC has the potential to outperform a type-1 FLC using a larger rule base; this can be shown in the control surface which graphically represent the unknown function articulated by the FLC controller. Figure 12.3a shows the control surface for the type-2 FLC steering output of the outdoor robot implementing the right edge following behavior using only two fuzzy sets to represent each input and thus having a rule base of four rules. This control surface represents the experiments shown in Figure 12.6b in which the outdoor robot will follow an irregular edge in an outdoor changing and dynamic environment. Note the smooth shape of the control surface which translates into a smooth control response that can deal with uncertainty and imprecision. Figure 12.3b shows the control surface for a type-1 FLC for the right edge following behavior also using two type-1 fuzzy sets to represent each input and thus having a rule base of four rules. Figure 12.4a shows the control surface for a type-1 FLC using three type-1 fuzzy sets to represent each input and thus having a rule base of eight rules. Figure 12.4b shows the control surface for a type-1 FLC using five type-1 fuzzy sets to represent each input and thus having a rule base of 25 rules. Note that the more type-1 fuzzy sets used in the type-1 FLC, the more its response approaches the smooth response of the type-2 FLC, as the type-2 fuzzy sets contain a large number of embedded type-1 sets which allow for the detailed description of the analytical control surface as the addition of the extra levels of classification gives a much smoother control surface and response [1].

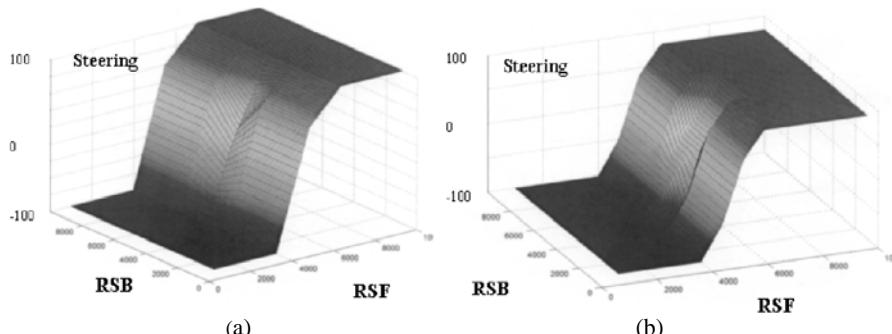
In the first part of our experiments, we used indoor robots navigating in noisy indoor unstructured environments. The robot path was drawn using a pen fixed to the back of the robot to record actual paths. In all the following experiments, all the average and standard deviations from the desired values were calculated over eight experiments using different setups and starting the robot from different locations.

Figure 12.5a shows the indoor robot implementing the type-2 FLC based left edge following behaviour to follow an irregular edge which offers poor ultrasound reflections. The robot is required to follow the edge at a desired distance of 35 cm. This type-2 FLC used a rule base of only four rules; it had succeeded in following the irregular edge with an average deviation from the desired distance of 1.6 cm and a standard deviation of 0.9. Figure 12.5a shows the type-2 FLC response which is a smooth response and can deal with the imprecision and uncertainty available in this indoor unstructured environment. We have compared the type-2 FLC response with type-1 FLCs using larger number of fuzzy sets (thus having larger rule bases).

Figure 12.5a shows the response of type-1 FLC which uses three type-1 fuzzy sets to represent each input; thus it has rule base of nine rules; the type-1 FLC had produced a path of average deviation from the desired distance of 3.1 cm and a standard deviation of 1.4. We experimented with another type-1 FLC with five type-1 fuzzy sets to represent each input and thus it has a rule base of 25 rules; the type-1 FLC had produced a path of average deviation of 2 cm from the desired distance and a standard deviation of 1.2. In these experiments the type-2 FLC had outperformed the type-1 FLC while using a smaller number of fuzzy sets to describe each input (thus having a smaller rule base). Also, as the number of type-1 fuzzy sets in type-1 FLC increases the type-1 performance approaches the type-2 performance which agrees with the control surface results in Figures 12.3 and 12.4



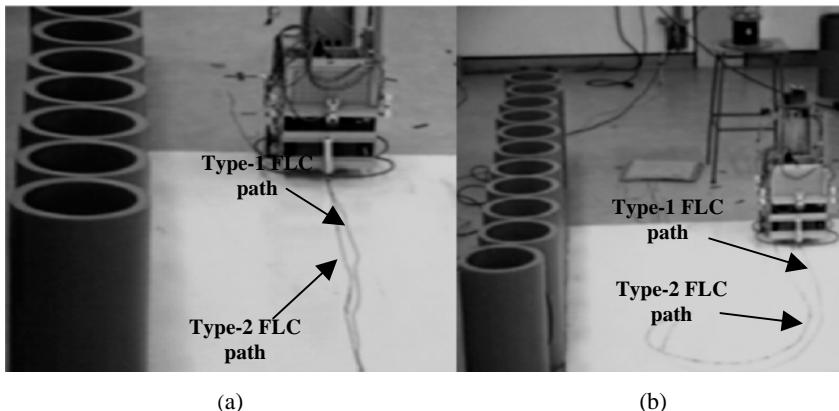
**Figure 12.3.** (a) The control surface of a type-2 FLC for the right edge following behaviour. (b) The control surface of a type-1 FLC using two fuzzy sets to represent each input



**Figure 12.4.** The control surface of a type-1 FLC: (a) Using three fuzzy sets; (b) Using five fuzzy sets

Figure 12.5b shows the indoor robot implementing the type-2 FLC-based goal seeking behaviour, where the goal is represented by an infrared beacon. The type-2 FLC for the goal seeking behaviour has a rule base of only three rules. Figure

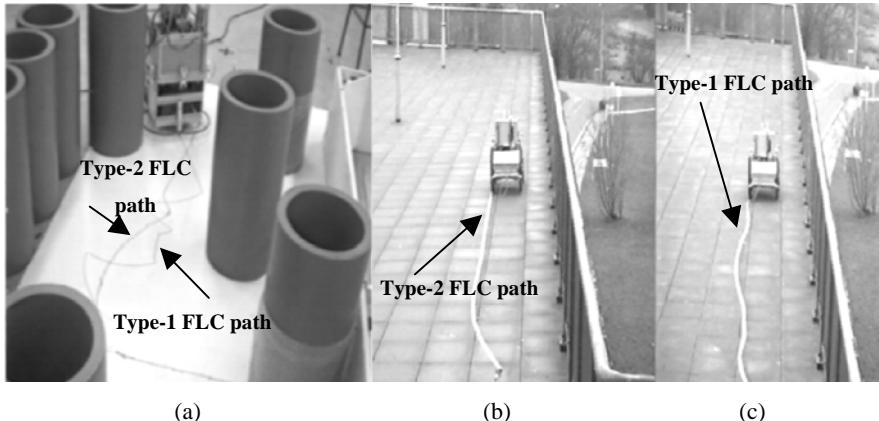
12.5b shows the path of the type-2 FLC where the goal is placed behind the robot and the robot turns and reaches the goal; the type-2 FLC had produced a smooth path with an average deviation of  $3^\circ$  and a standard deviation of 0.7 (from the desired zero degree alignment). We have tried type-1 FLCs using more fuzzy sets to represent each input (thus having larger rule bases). Figure 12.5b shows a type-1 FLC using seven type-1 fuzzy sets to represent the input and thus having a rule base of seven rules; this type-1 FLC has given larger values for the average and the standard deviations.



**Figure 12.5.** (a) Indoor robot path using type-2 FLC to implement the left edge following behaviour compared against type-1 FLC path using three fuzzy sets to represent each input. (b) The robot path using type-2 FLC to implement the goal seeking behaviour compared against a type-1 FLC using seven fuzzy sets

Figure 12.6a shows the robot using a type-2 FLC to perform the obstacle avoidance behaviour; the robot goal is to maintain a minimum safe distance of 45 cm from any front obstacle. The type-2 FLC for the obstacle behaviour has a rule base of only eight rules. Figure 12.6a shows the path of the type-2 FLC where the robot navigates in a tight environment; the type-2 FLC had produced a smooth path with an average deviation from the desired safe distance of 2.4 cm and a standard deviation of 1.1. We have tried type-1 FLCs using a higher number of fuzzy sets to represent each input (thus having larger rule bases). Figure 12.6a shows the path of a type-1 FLC having a rule base of 27 rules; this type-1 FLC had given an average deviation of 5.3 cm and a standard deviation of 3.2.

Although we did what we could to introduce as much noise, imprecision and dynamic changes to the indoor environments to evaluate the real time performance of the type-2 FLC, there are clearly big differences between indoor environments and the outdoor changing and dynamic unstructured environments. The outdoor unstructured environments will be a severe test to evaluate the real time performance of the robot type-2 FLC and how it handles the large amounts of uncertainty and imprecision facing mobile robots.



**Figure 12.6.** (a) The type-2 FLC implementing the obstacle avoidance behaviour compared with a type-1 FLC using three fuzzy sets to represent each input. (b) The outdoor robot path using type-2 FLC to implement the right edge following behaviour. (c) The robot path using a type-1 FLC that gave a bad response when the environment changed (windy weather)

We had performed many experiments using our outdoor robots in outdoor changing and dynamic environments. The robot path was recorded using a tape fixed under the left back wheel and we calculated the average and standard deviations from the desired values. In all the following experiments, all the average and standard deviations from the desired values were calculated over eight experiments where the robot was started from different random locations and under different environmental conditions (such as wind, rain, sunshine, *etc.*) and in different ground conditions (slippery, dry grounds). Figure 12.6b shows the outdoor robot implementing the type-2 based right edge following behaviour to follow an irregular metallic edge at a desired distance of 1.2 m. The type-2 FLC has four rules in the rule base. The robot type-2 FLC had followed the irregular edge with an average deviation of 2.1 cm and a standard deviation of 0.8. The type-1 FLC can give a good response at a specific weather, ground and robot condition but if any of these conditions change like operating in a windy weather condition then the type-1 FLC with nine rules shown in Figure 12.6c will fail and give a bad response as it is using crisp and precise membership functions that cannot handle the uncertainties associated with changing unstructured outdoor environments, while the type-2 FLC can handle such uncertainties to give a better response while using a smaller rule base.

## 12.5 Conclusions

In this chapter, we presented the interval type-2 FLC and its novel application to mobile robots real time control in changing and dynamic unstructured indoor and outdoor environments. To the author's knowledge, this is the first work applying type-2 fuzzy systems to real time robot control and to real time control in general.

We have used the type-2 FLC to implement different mobile robot behaviours in different platforms in indoor and outdoor challenging unstructured environments. We have shown how the type-2 FLCs can deal with the uncertainties facing mobile robots in unstructured environments to produce very good real time responses that outperformed the type-1 FLCs whilst using smaller rule bases than the type-1 FLCs. Based on these results, we are exploring the application of type-2 FLCs to other control application such as the control of huge marine diesel engines and intelligent buildings. We are also exploring automatic learning techniques to learn the various parameters of the type-2 FLC online.

## References

- [1] A. Kandel and G. Langholz, *Fuzzy Control Systems*, Boca Raton : CRC Press, 1994.
- [2] N. Karnik, J. Mendel and Q. Liang, "Type-2 Fuzzy logic Systems", *IEEE Transactions on Fuzzy Systems*, Vol.7, December 1999, pp 643-658.
- [3] G. Klir and T. Fogler, *Fuzzy Sets, Uncertainty and Information*. Prentice Hall, 1988
- [4] Q. Liang, N. Karnik and J. Mendel, "Connection Admission Control in ATM Networks using Survey-Based Type-2 Fuzzy Logic Systems", *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, Vol.30, August 2000, pp 329-339.
- [5] Q. Liang, and J. Mendel, "Interval Type-2 Fuzzy Logic Systems: Theory and Design", *IEEE Transactions on Fuzzy Systems*, Vol. 8, October 2000, pp 535-550.
- [6] C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II", *IEEE Transactions on Systems, Man and Cybernetics*, Vol.20, March 1990, pp. 419-434.
- [7] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Upper Saddle River, NJ: Prentice-Hall, 2001.
- [8] J. Mendel and R. John, "Type-2 Fuzzy Sets Made Simple", *IEEE Transactions on Fuzzy Systems*, Vol.10, April 2002, pp 117-127.
- [9] H. Hagras, "A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots", *IEEE Transactions on Fuzzy Systems*, Vol.12, August 2004, pp 524-539.
- [10] A. Saffiotti, "The Uses of Fuzzy Logic in Autonomous Robot Navigation", *Journal of Soft Computing*, Vol.1, No.4, 1997, pp 180-197.
- [11] A. Saffiotti, "Fuzzy Logic in Autonomous Robotics: Behaviour Coordination", *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, Barcelona, Spain, 1997, pp 573-578.
- [12] E. Tunstel, T. Lippincott and M. Jamshidi, "Behaviour Hierarchy for Autonomous Mobile Robots: Fuzzy Behaviour Modulation and Evolution", *International Journal of Intelligent Automation and soft computing*, Vol.3, No.1, 1997, pp 37-49.
- [13] K. Wu, "Fuzzy Interval Control of Mobile Robots", *Journal of Computers Electrical Engineering*, Vol. 22, No.3, 1996, pp 211-229.
- [14] H. Wu and J. Mendel, "Uncertainty Bounds and their Use in the Design of Interval Type-2 Fuzzy Logic Systems", *IEEE Transaction on Fuzzy Systems*, Vol.10, No.5, October 2002, pp 622-639.

## Combine Sliding Mode Control and Fuzzy Logic Control for Autonomous Underwater Vehicles

Feijun Song and Samuel M. Smith

### 13.1 Introduction

Autonomous Underwater Vehicle (AUV) is an emerging and proven technology by which four-dimensional synoptic data can be collected and rapid environment assessment becomes realizable. Low level control, for example pitch and heading control, is the basis for the successful accomplishment of any mission carried out by an AUV. Many control strategies have been applied to Autonomous Underwater Vehicles; among them are neural network control [1], sliding mode control [2, 3], supervisory control [4], linear quadratic Gaussian / loop transfer recovery method [5], self-tuning control [6], fuzzy logic control [7], *etc.* However, due to the complexity of an AUV system, it is difficult to design a nonlinear controller for AUVs with good performance.

Fuzzy logic control and sliding mode control have been developed and applied to nonlinear system for the last two decades [8, 9]. The most attractive feature of fuzzy logic control is that expert knowledge can be easily incorporated into the control laws. Although a Fuzzy Logic Controller (FLC) is similar to a Sliding Mode Controller (SMC) [10], the combination of fuzzy logic control and sliding mode control has still been of research interest due to the fact that the stability of a general fuzzy logic controller is difficult to prove whereas the stability of a sliding mode controller is inherent. Other benefits of sliding mode control are its invariance property and the ability to decouple high dimensional systems into subsystems with lower dimensionality [8]. This feature can help to reduce the rule base size of an FLC. Another direct benefit of combining fuzzy logic control and sliding mode control is that fuzzy logic can effectively eliminate chattering, which is a common problem in sliding mode control.

In a Takagi-Sugeno (TS) type FLC [10], the rule output function is typically a liner function of controller inputs. The mathematical expression of this function is similar to a switching function. Wu and Liu proposed a sliding mode approach to fuzzy controller design [11]. In their approach, parameters in the output function are determined according to sliding mode control theory. Every parameter accordingly has its own physical meaning as those in sliding mode theory. The resulting controller is still a typical TS type.

However most work on combining fuzzy logic and sliding mode theory has been done in an opposite direction, *i.e.*, using fuzzy logic to help design a sliding mode controller. In a SMC, the sliding region is a hyperplane [8]. In the case of two-dimensional systems, the sliding region will be just a line. The control action will be maximal on one side of the line and be minimal on the other side. Theoretically, it is assumed that a controller can switch from one extreme to another extreme infinitely fast. A system trajectory then can stay on the sliding line as soon as it reaches the line. However in practice, due to limited sampling rate, system trajectories chatter instead of sliding along the switching line. The control action alternatively changes its sign along the switching line.

To eliminate the chattering, a boundary layer can be introduced around the sliding surface and the tracking performance is compromised [12]. Within the boundary layer, the control command is a linear function of the distance between the actual system state and the sliding surface. The distance is represented by a function called sliding function.

Fuzzy logic is being used to map the distance to the control command as a nonlinear Single-Input-Single-Output function. Lim proposed a control scheme that combines a Proportional Integral Derivative (PID) and a fuzzy logic controller together. The error, error rate and the error integral are first combined together linearly as those in a PID controller, then the scalar result is sent to a single input single output FLC [13]. The coefficients for the PID controller have to be pre-selected. This actually is a fuzzy approach to the sliding mode controller design. Multiple input has been combined as a single input. The linear combination of error, error rate and error integral is actually the distance of the actual system state and the sliding surface. Lim's approach is similar to that of Kung's [14]. Other researchers have taken a similar approach [15-17].

Wong proposed a combination of a sliding mode controller and a PI controller using an FLC to eliminate the steady state error for type-0 systems. The stability of the resulting controller is proved through a Lyapunov function [18].

Lin proposed a Genetic Algorithm based controller design approach [19]. Lin first fuzzified a crisp sliding surface into a fuzzy sliding surface to reduce the chattering, then used GA to generate the membership functions of the consequent fuzzy sets. The fuzzy controller is still a one input one output fuzzy controller that fuzzifies the sliding surface. His approach is very similar to that of Ng [20].

Hsu also proposed a similar approach; however he included the changes of sliding surface into the controller so the fuzzy controller is now a two input one output controller [21]. Hsu's approach is very similar to those of Li and Allamehzadeh [22,23] and is exactly the same as that of Hwang [24].

In all those approaches, a simple FLC is used to fuzzify the relationship of the control command and the distance between the actual state and the sliding surface. This kind of approach could be called Fuzzy Sliding Mode Controller (FSMC) since fuzzy logic is used only to fuzzify a sliding surface. After fuzzification, the sliding surface is not a hyperplane any more. In the 2D case, the sliding line becomes a band of sliding area. This in essence is equal to introducing a boundary layer; however the control command changes nonlinearly inside the band. A boundary layer decreases the controller performance. Some researchers proposed

different fuzzy variable boundary layer approaches to alleviate the problem [25,26].

Similar to gain scheduling, since different gains should be used in different regions of a state space, the coefficients of the sliding hyperplane should also be different in different regions of the state space to improve the transient response. All the above approaches never address this problem. Boverie proposed a scheme that replaces the linear sliding surface with a piecewise nonlinear surface [27]. Smith adaptively changed the coefficients of a sliding surface according to the overshoot of a step response by changed the input output scaling factors [7]. Xu took an even more direct approach [28]. In Xu's approach, each rule is a sliding mode controller. The boundary layer and the coefficients of the sliding surface become the coefficients of the rule output function and have their physical meanings. Xu's approach makes the adjustment of the boundary layer and the sliding surface easier and more directly. Xu's approach results in a Sliding Mode Fuzzy Controller that is different from a FLC tuned by sliding mode theory.

This chapter first presents the general background on an AUV system, followed by general sliding mode control theory. Then a detailed analysis of Fuzzy Sliding Mode Controllers (FSMC) and Sliding Mode Fuzzy Controllers (SMFC) is presented. The last section of this chapter is the sea trial data for the SMFC designed for an AUV system.

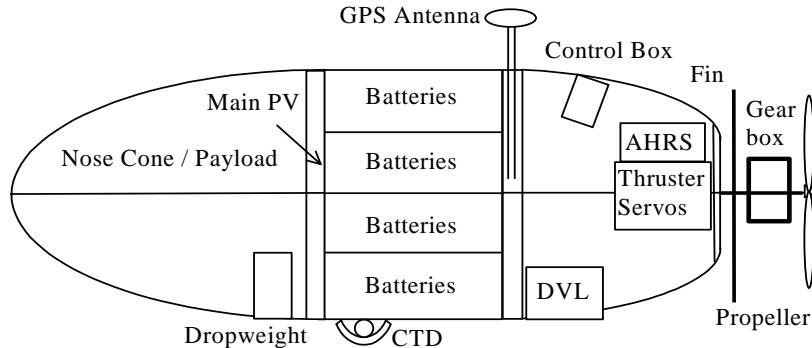
## 13.2 Autonomous Underwater Vehicles (AUVs)

OEX series AUVs were developed at the Ocean Engineering Department of Florida Atlantic University. The vehicle is depicted in Figure 13.1. It is *7.14 feet* long with basic payload and *21 inches* in diameter. The maximum cross sectional area is *2.4053 ft<sup>2</sup>*. Weight in air is *714.2 lbs* and displaced weight is *716.7 lbs*. Hull volume is *11.1931 ft<sup>3</sup>*. Following is a brief description of the basic vehicle configuration:

- A tear-drop shaped fiber glass hull based on a modified version of the Gertler Series 58 Model 4154 body shape
- Aft-mounted cruciform control surfaces
- A three-bladed propeller *18 inches* in diameter
- Intelligent Ni-Cd battery packs. The battery packs can supply up to *12 hours* of continuous missions at three-knot cruising speed
- Main computer and electronics board (MC68030 at 50 MHz on the VME bus). Each of the components is embedded with a LonWorks Neuron node, and the control communication is achieved via LonTalk protocol
- Sensors include Watson AHRS-C302RS (three-axis acceleration, angles and rates), SIMRAD mesotech 809 (altitude), Druck PTX 1649 (water depth), Sonic Speed (water speed), Differential Global Positioning System, LBL and USBL positioning system

In OEX series AUVs, the control of heading and pitch is achieved through the adjustment of rudder and stern plane. We will design two time optimal controllers

for pitch and heading control of OEX series AUVs. The inputs for the pitch controller are the pitch error and pitch error rate. The output of the pitch controller is the deflection of the stern plane. The inputs for the heading controller are heading error and heading error rate. The output of the heading controller is the deflection of the rudder.



**Figure 13.1.** OEX series AUVs.

### 13.3 Sliding Mode Controller

A Sliding Mode Controller is a Variable Structure Controller (VSC). Basically, a VSC includes several different continuous functions that map plant state to a control surface, and the switching among different functions is determined by plant state that is represented by a switching function.

Without loss of generality, consider the design of a sliding mode controller for the following second order system:

$$\ddot{x} = f(x, \dot{x}, t) + bu(t)$$

Here we assume  $b \neq 0$ .  $u(t)$  is the input to the system. The following is a possible choice of the structure of a sliding mode controller [2]:

$$u = -k \operatorname{sgn}(s) + u_{eq} \quad (13.1)$$

where  $u_{eq}$  is called equivalent control which is used when the system state is in the sliding mode [8].  $k$  is a constant, representing the maximum controller output.  $s$  is called switching function because the control action switches its sign on the two sides of the switching surface  $s = 0$ .  $s$  is defined as [9]

$$s = \dot{e} + \lambda e \quad (13.2)$$

where  $e = x - x_d$  and  $x_d$  is the desired state.  $\lambda$  is a constant. The definition of  $e$  here requires that  $k$  in Equation (13.1) be positive.  $\text{sgn}(s)$  is a sign function, which is defined as

$$\text{sgn}(s) = \begin{cases} -1 & \text{if } s < 0 \\ 1 & \text{if } s \neq 0 \end{cases}$$

The control strategy adopted here will guarantee a system trajectory moves toward and stays on the sliding surface  $s = 0$  from any initial condition if the following condition applies:

$$s\dot{s} \leq -\eta|s|$$

where  $\eta$  is a positive constant that guarantees the system trajectories hit the sliding surface in finite time [9]

Using a sign function often causes chattering in practice. One solution is to introduce a boundary layer around the switch surface [9]:

$$u = -ksat\left(\frac{s}{\phi}\right) + u_{eq} \quad (13.3)$$

where constant factor  $\phi$  defines the thickness of the boundary layer.  $sat\left(\frac{s}{\phi}\right)$  is a saturation function that is defined as

$$sat\left(\frac{s}{\phi}\right) = \begin{cases} \frac{s}{\phi} & \text{if } \left|\frac{s}{\phi}\right| \leq 1 \\ \text{sgn}\left(\frac{s}{\phi}\right) & \text{if } \left|\frac{s}{\phi}\right| > 1 \end{cases}$$

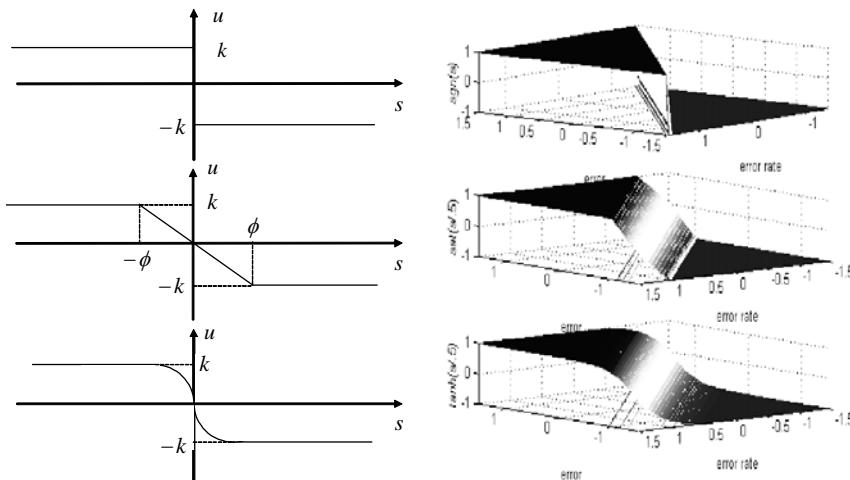
This controller is actually a continuous approximation of the ideal relay control [9]. The consequence of this control scheme is that invariance property of sliding mode control is lost. The system robustness is a function of the width of the boundary layer.

A variation of the above controller structures is to use a hyperbolic tangent function instead of a saturation function [9]:

$$u = k \tanh\left(\frac{s}{\phi}\right) + u_{eq} \quad (13.4)$$

It is proven that if  $k$  is large enough, the sliding model controllers of Equations (13.1), (13.3) and (13.4) are guaranteed to be asymptotically stable [9].

For a two-dimensional system, the controller structure and the corresponding control surface are illustrated in Figure 13.2.



**Figure 13.2.** Various sliding mode controllers and control surfaces

### 13.4 Fuzzy Sliding Mode Control (FSMC)

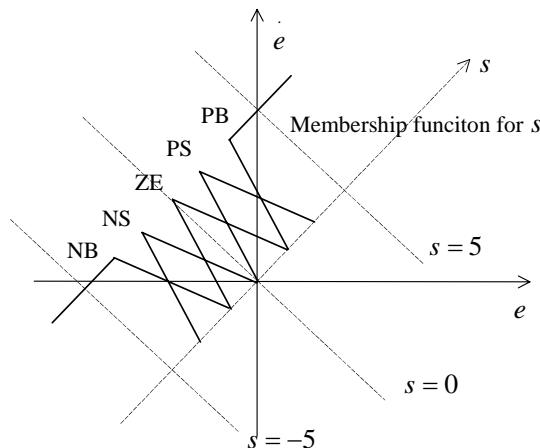
As mentioned before, in a Sliding Mode Controller, the sliding surface is a hyperplane [8]. In the case of two-dimensional systems, the sliding surface will be just a line without any width. Control action then alternately changes its sign when the system state switches from one side of the line to the other side of the line as the line has no width. A boundary layer can alleviate the chattering problem.

Most of the research effort on combining fuzzy logic control and sliding mode control has been spent on how to use fuzzy logic to approximate the control command  $u$  as a nonlinear function of  $s$  within the boundary layer [13-25]. The resulting Fuzzy Sliding Mode Controller (FSMC) is actually a Single-Input-Single-Output fuzzy logic controller. The input to the controller is the switching function  $s$ . The output from the controller is the control command  $u$ . The number of rules in the rule base depends on the fuzzification level of  $s$  and  $u$ ; however it is generally much less than a typical fuzzy logic controller for the same system. A typical rule of a FSMC has the following format:

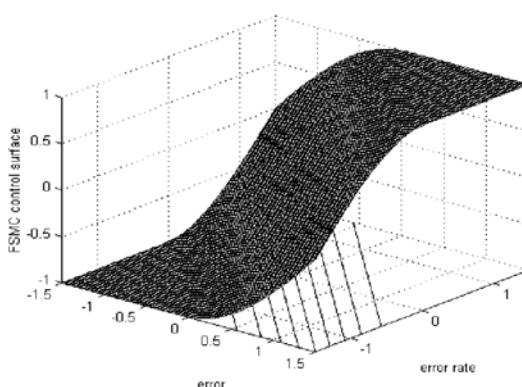
IF  $s$  is PB (and  $\dot{s}$  is PB), THEN  $u$  is PB or a constant.

How the switching function  $s$  is fuzzified is illustrated in Figure 13.3. This kind of rule format effectively adds a boundary layer to the system. This boundary layer is illustrated in Figure 13.4. Figure 13.5 illustrates the relationship of control command  $u$  and switching function  $s$ .

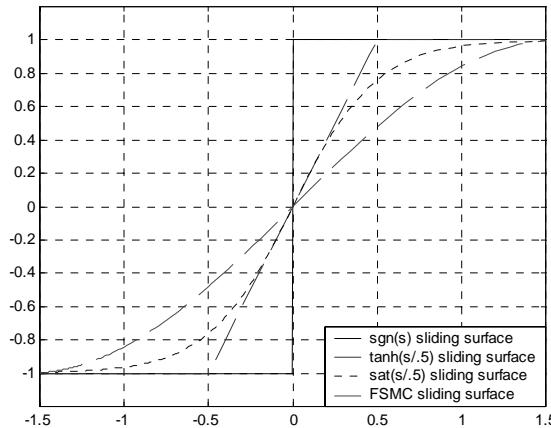
The main drawback of an FSMC is that, before the fuzzification, the coefficients of the sliding surface have to be pre-defined carefully by an expert. That is, the value of  $\lambda$  in Equation (13.2) has to be determined first. The only advantage of an FSMC compared with a typical SMC is that in the former case, the boundary layer is nonlinear whereas in the latter case the boundary layer is linear. A nonlinear boundary does not add much control performance since it is the width of the boundary layer that influences the controller performance, not the form of the boundary layer [9]. From authors' point of view, FSMC has little practical value, if not being totally useless.



**Figure 13.3.** The fuzzification of switching function  $s$  for a fuzzy sliding mode controller



**Figure 13.4.** The control surface and the effective boundary layer of a fuzzy sliding mode controller

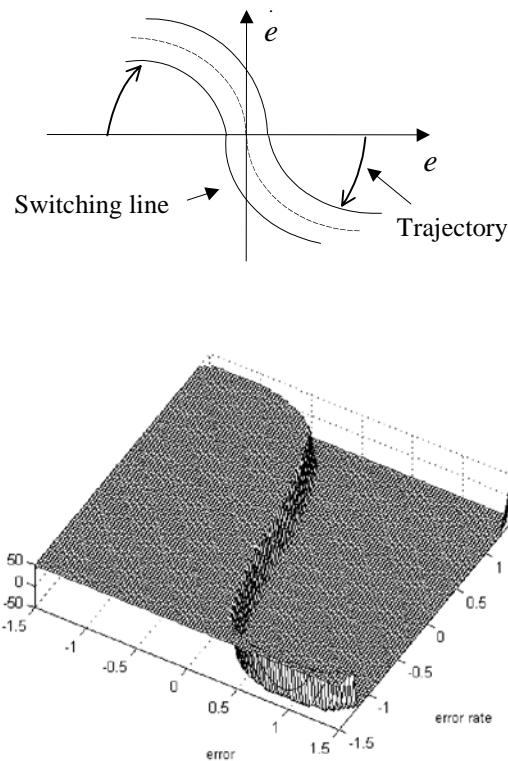


**Figure 13.5.** The relationship of control command  $u$  and switching function  $S$

### 13.5 Sliding Mode Fuzzy Control (SMFC)

Sliding mode control is a kind of time sub-optimal control. The well-known solution for time optimal set point control of a system with bounded control action is a bang-bang controller across a nonlinear switching curve. For two-dimensional systems, the nonlinear switching curve often has the form depicted in Figure 13.7 instead of a linear switching line. Figure 13.6 also shows there can be a switching band around the switching line to alleviate chattering. Figure 13.4 also illustrates how the control surface may look with a nonlinear switch function.

In a Takagi-Sugeno (TS) type FLC, the rule output function typically is a linear function of controller inputs. The mathematical expression of this function is similar to a switching function. This similarity indicates that the information from a sliding mode controller can be used to design a fuzzy logic controller, resulting in a Sliding Mode Fuzzy Controller (SMFC). Wu proposed such an approach in which parameters in the output functions for different rules that covers different partitions of the state space are determined by different sliding mode controllers that also cover the corresponding partitions of the state space [11]. The resulting controller is still a typical TS type FLC. In fact, since a fuzzy system can seamlessly connect different control strategies into one system, one can take an even more direct approach to incorporate sliding mode controllers into a fuzzy logic controller [28]. In Xu's approach, each rule is a sliding mode controller. The SMC in each rule can have various forms. The boundary layer and the coefficients of the sliding surface become the coefficients of the rule output function and have their physical meanings.

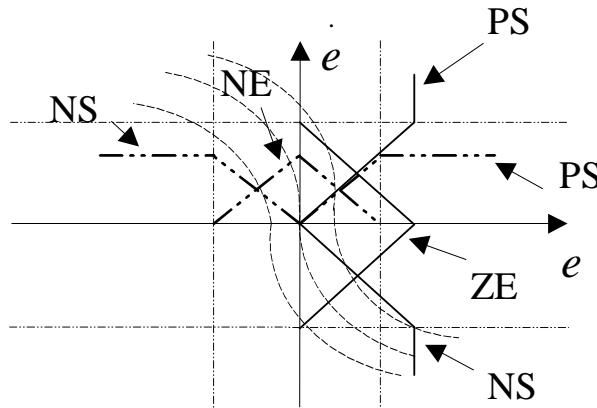


**Figure 13.6.** A nonlinear switching curve and its corresponding control surface

The  $i$  th rule for a Sliding Mode Fuzzy Controller is expressed as follows:

$$\text{IF } e \text{ is } A_i \text{ and } \dot{e} \text{ is } B_i, \text{ THEN } u_i = \text{ksat}\left(\frac{\dot{e} + \lambda_i e + c_i}{\phi_i}\right)$$

Notice that there are still as many inputs to the fuzzy controller as the number of state variables. The structure of the fuzzy controller does not reduce the rule base size. However, this kind of SMFC can achieve more complicated switching functions with less rules than an ordinary FLC does. Also notice that the rule output function is not necessarily a saturation function; it could be a *Sign* function or hyperbolic tangential function. The fuzzification of  $e$  and  $\dot{e}$  are illustrated in Figure 13.7.



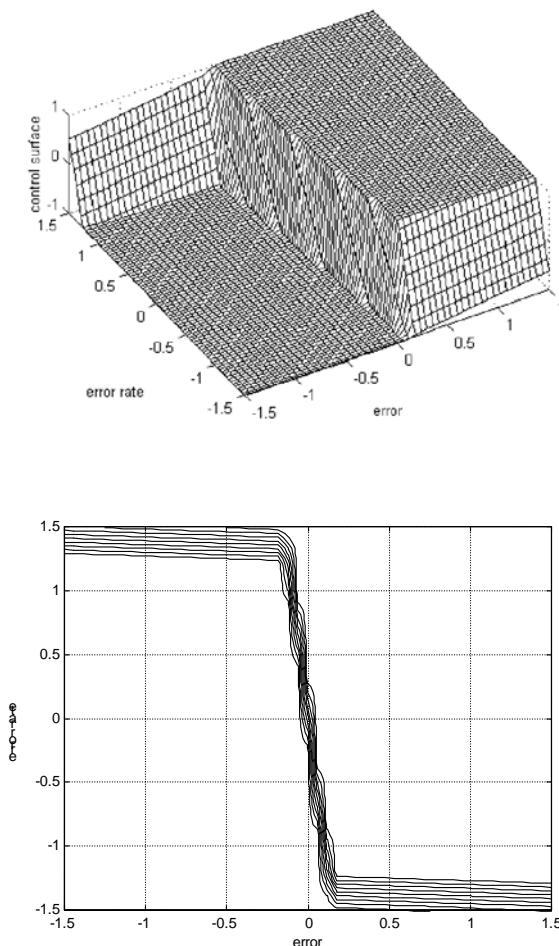
**Figure 13.7.** Fuzzification of  $e$  and  $\dot{e}$

An FSMC is intended to approximate a nonlinear sliding surface, not a nonlinear switching function. Within an FSMC, the switching function is still a linear function as in Equation (13.2). However, an SMFC can approximate a nonlinear switching function easily with just a few rules. Moreover, for a two-dimensional system, the switching line can be either a function of  $e$ , or a function of  $\dot{e}$  since it is a one-dimensional function. A much smaller number of rules are needed to approximate this one-dimensional function. This is how a Sliding Mode Fuzzy Controller reduces the rule base size.

To illustrate how the rule base size is reduced, we use a two-dimensional system. Under the typical Sliding Mode Fuzzy Controller, with each input fuzzified into three fuzzy subsets as shown in Figure 13.8, there are nine partitions of the state space, and nine rules are needed. However, with the following simplified rule:

$$\text{IF } e \text{ is } A_i, \text{ THEN } u_i = \text{ksat}\left(\frac{\dot{e} + \lambda_i e + c_i}{\phi_i}\right)$$

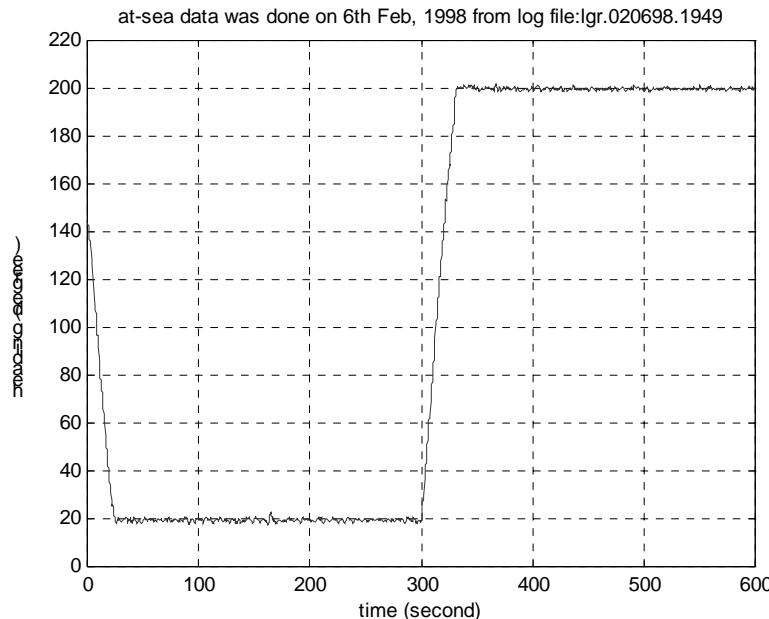
The rule base size may be less than nine. To approximate the switching curve in Figure 13.7,  $e$  may need more fuzzy subsets. It has been shown that five membership functions for  $e$  are enough to approximate the switching curve in Figure 13.7. The control surface of such a five rule FSMC and the corresponding sliding surface is shown in Figure 13.8.



**Figure 13.8.** The control surface and sliding region for a five-rule SMFC

### 13.6 Sliding Mode Fuzzy Control Applied to AUVs

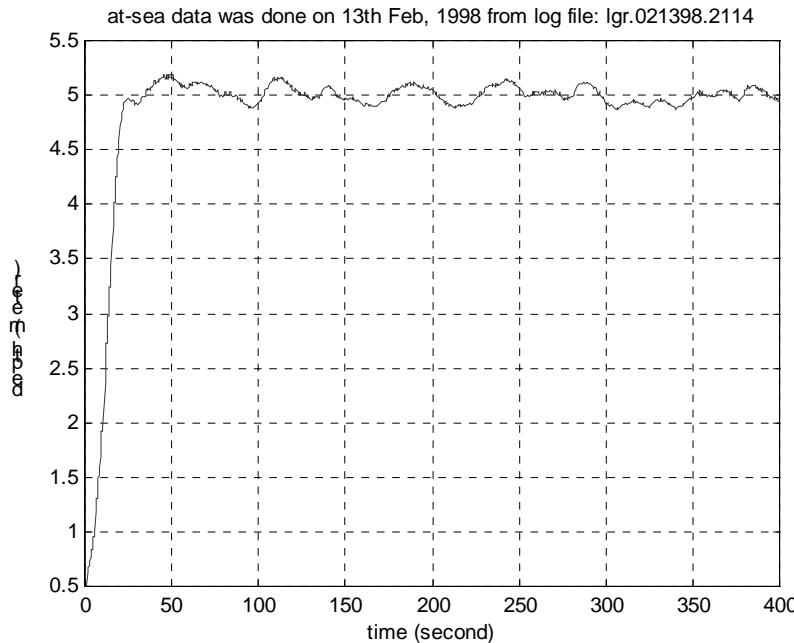
A sliding mode fuzzy pitch controller and a sliding mode fuzzy heading controller have been designed for the OEX series AUVs. The inputs to the sliding mode fuzzy heading controller are heading error and heading error rate. The output is rudder deflection. The inputs to the pitch controller are pitch error and pitch error rate. The output is stern plane deflection.



**Figure 13.9.** At-sea experiment data for heading control

Figure 13.9 shows the performance of the heading controller in open water trial. The commanded heading is first set to  $20^\circ$  and then  $200^\circ$ . The heading controller successfully controlled the vehicle to the desired heading. There is a  $2$  degrees oscillation in the at-sea heading. This is normal since the sea environment is not as clean as the simulation environment.

Figure 13.10 shows the performance of the pitch controller in at-sea test. Since in the experiment, the vehicle cannot have a constant pitch with limited sea depth, the performance of the pitch controller was tested through the depth control. In OEX series AUVs, the depth control is done through the pitch control. The control output from the depth controller is a desired pitch angle. It is the responsibility of the pitch controller to drive the vehicle to the desired pitch. In Figure 13.10, the depth command is set to *5 meter*. The depth controller is a simple PD controller. Together with the pitch controller, it successfully drives the vehicle to *5-meter* depth.



**Figure 13.10.** At-sea experiment data for depth control

## 13.7 Conclusions

In this chapter, fuzzy sliding mode controllers and sliding mode fuzzy controllers were distinguished, and their respective properties were discussed. A sliding mode fuzzy controller inherits the interpolation property of fuzzy logic control and robustness property of sliding mode control, thereby making it ideal for time optimal robust control. The at-sea experimental data shows that sliding mode fuzzy control can be successfully applied to AUVs, thereby making them another alternative for the robust time optimal control problem. Moreover, since the physical meaning of the rule output function parameters for a sliding mode fuzzy controller is straightforward, the online tuning is made easy. This is exactly why sliding mode fuzzy control is adopted in AUV development at Florida Atlantic University.

## References

- [1] J. Yuh, "A Neural Net Controller For Underwater Robotic Vehicles," *IEEE Journal of Oceanic Engineering*, Vol. 15, No. 3, pp 161-166, 1990.
- [2] D.R. Yoerger and Jean-Jacques E. Slotine, "Robust Trajectory Control of Underwater Vehicles," *IEEE Journal of Oceanic Engineering*, Vol. 10, No. 4, pp 462-470, 1985.

- [3] A.J. Healey and D. Lienard, "Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles," *IEEE Journal of Oceanic Engineering*, Vol. 18, No. 3, 1993.
- [4] D.R. Yoerger, J.B. Newman and Jean-Jacques E. Slotine, "Supervisory Control System for the JASON ROV," *IEEE Journal of Oceanic Engineering*, Vol. 11, No. 3, pp 392-399, 1986.
- [5] M.S. Triantafyllou and M.A. Grosenbaugh, "Robust Control for Underwater Vehicle Systems with Time Delays," *IEEE Journal of Oceanic Engineering*, Vol. 16, No. 1, pp 146-151, 1991.
- [6] K.R. Goheen and E.R. Jefferys, "Multivariable Self-Tuning Autopilots for Autonomous and Remotely Operated Underwater Vehicles," *IEEE Journal of Oceanic Engineering*, Vol. 15, No. 3, pp 144-151, 1990.
- [7] S.M. Smith, "A Variable-Structure Fuzzy Logic Controller with Run-time Adaptation," *IEEE International Conference on Fuzzy Systems*, pp 983-988, 1994.
- [8] V.I. Utkin, "Sliding Modes in Control and Optimization," *Springer-Verlag*, 1992.
- [9] John Y. Hung, W. Gao and James C. Hung, "Variable Structure Control: A Survey," *IEEE Transactions on Industrial Electronics*, Vol. 40, No. 1, pp 2-21, 1993.
- [10] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Transactions on System, Man, and Cybernetics*, Vol.15, pp 116-132, 1985.
- [11] J.C. Wu and T.S. Liu, "A Sliding-Mode Approach to Fuzzy Control Design," *IEEE Transaction on Control Systems Technology*, Vol. 4, No. 2, pp 141-150, 1996.
- [12] J.J. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [13] S.C. Lin and C.C. Kung, "Linguistic Fuzzy-Sliding Mode Controller," *American Control Conference*, pp 1904-1905, 1992.
- [14] C.M. Lim, "Application of a New Fuzzy Logic Control Scheme for Stability Enhancement - Simulation and Experimental Results," *IEEE International Conference on Fuzzy Systems*, pp 1339-1346, 1995.
- [15] C.C. Kung and S.C. Lin, "A Fuzzy-Sliding Mode Controller Design," *IEEE International Conference on Systems Engineering*, pp 608-611, 1992.
- [16] N.C.M. Leung and C.K. Li, "A Multi-Phase Fuzzy Variable Structure Controller with Improved Transient and Steady State Response," *IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, pp 262-266, 1993.
- [17] J.S. Glower and J. Munighan, "Designing Fuzzy Controllers from a Variable Structures Standpoint," *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 1, pp 138-144, 1997.
- [18] L.K. Wong, F.H.F. Leung and P.K.S. Tam, "Combination of Sliding Mode Controller and PI Controller Using Fuzzy Logic Controller," *IEEE International Conference on Fuzzy Systems*, pp 296-301, 1998.
- [19] Sinn-Cheng Lin and Yung-Yaw Chen, "A GA-Based Fuzzy Controller with Sliding Mode," *IEEE International Conference on Fuzzy Systems*, pp 1103-1110, 1995.
- [20] K.C. Ng, Y. Li, D.J. Murray-Smith and K.C. Sharman, "Genetic Algorithms Applied to Fuzzy Sliding Controller Design," *IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp.220-225, 1995.
- [21] Ya-Chen Hsu and H.A. Malki, "Fuzzy Variable Structure Control for MIMO Systems," *IEEE International Conference on Fuzzy Systems*, pp 280-285, 1998.
- [22] Tzuu-Hseng S. Li and Chin-Yin Tsai, "Parallel Fuzzy Sliding Mode Control of the Cart-Pole System," *IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, pp 1468-1473, 1995.

- [23] H. Allamehzadeh and J.Y. Cheung, "Design of a Stable and Robust Fuzzy Controller for a Class of Nonlinear System," *IEEE International Conference on Fuzzy Systems*, pp 2150-2154, 1996.
- [24] Guang-Chyan Hwang and Shih-Chang Lin, "A Stability Approach to Fuzzy Control Design for Nonlinear System," *Fuzzy Sets and Systems*, Vol. 48, pp 279-287, 1992.
- [25] Yean-Ren Hwang and M. Tomizuka, "Fuzzy Smoothing Algorithm for Variable Structure Systems," *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 4, pp 277-284, 1994.
- [26] H. Lee, H. Son, E. Kim and M. Park, "Variable Structure Control of Nonlinear System Using Fuzzy Variable Boundary Layer," *IEEE International Conference on Fuzzy Systems*, pp 348-353, 1998.
- [27] S. Boverie, P. Cerf and J.M. Le Quellec, "Fuzzy Sliding Mode Control Application to Idle Speed Control," *IEEE International Conference on Fuzzy Systems*, pp 974-977, 1994.
- [28] M. Xu, "High Performance and Robust Control," *Ph.D. Dissertation, Florida Atlantic University*, 1996.

## Fuzzy Logic for Flight Control I: Nonlinear Optimal Control of Helicopter Using Fuzzy Gain Scheduling

Gwo-Ruey Yu

### 14.1 Introduction

Helicopters exhibit high levels of agility and maneuverability, performing several actions, such as “climbing”, “hovering”, and “forward flight” [1]. A helicopter with superior performance is needed. With its high agility and performance, a helicopter is also unstable exhibiting non-linear qualities. In view of this, a more effective helicopter flight control strategy is developed in this study to increase the stability and performance. Linear optimal controllers are designed in the flight envelope operating points. The Takagi-Sugeno (TS) fuzzy gain-scheduler is implemented to accomplish the desired specifications.

Linear quadratic (LQ) optimization has been utilized extensively in the field of aerospace engineering. LQ technology designs an optimal controller that minimizes a given performance index. The performance index is parameterized using state-weighting matrix  $Q$  and control-weighting matrix  $R$ . The advantage of LQ methodology is that it easily obtains the optimal gains for the various linear models used in gain scheduling [2]. Moreover, the optimal gains have gain margins from 0.5 to infinity and phase margins of  $60^\circ$  or more [3]. Another benefit is that this controller produces constant gain in most cases [4]. After solving the Riccati equation, the linear optimal controller is obtained for the linearized helicopter model around an equilibrium point.

A challenging task for control engineers is helicopter autopilot designs that achieve the desired robust stability and performance over a large flight envelope (defined as altitude *vs* Mach number). A typical control design method employed for helicopters operating over a large flight envelope is gain scheduling. There is no general procedure for the interpolation of the already available control law into a gain-scheduler. However, it is effective for gain-scheduler design using fuzzy logic [5]. L.A. Zadeh first proposed the fuzzy set idea. The fuzzy inference system employing fuzzy if-then rules can control a plant using human knowledge. Fuzzy controllers have been widely used in industry because of their easy realization and robustness [6, 7]. Much work has been done on the analysis of control rules and membership function parameters. The TS fuzzy model can be regarded as a particular case of the general fuzzy model. The consequence of the TS model is a

linear combination of the linguistic variables in antecedents. The output of a zero-order TS model is a singleton only [8].

Two types of fuzzy controllers are combined to control an autonomous helicopter [9]. The Mamdani-type controller regulates the desired velocity and the TS-type controller achieves the desired attitude angle. A fuzzy gain-scheduler design for tracking the reference trajectory of a robot is presented in [10]. The “rules evolution tuning method” and “knowledge based adjustment” are devised for fuzzy control of an unmanned helicopter [11]. A comparison of fuzzy-logic, LQR and neural-net controllers is proposed for a UH-1H helicopter [12]. The fuzzy-logic controller is superior to the LQ regulator under severe gust turbulence. The literature [13] utilizes fuzzy logic and a genetic algorithm UH-1H helicopter control design. The genetic algorithm is applied to search for the optimal rules in a fuzzy inference system. The gain-scheduler, based on LQ linear controller, is implemented for a miniature acrobatic helicopter [14]. However, the LQ feedback gains are only linearly interpolated.

In this chapter, an LQ optimal controller design with fuzzy gain-scheduler applied to nonlinear Bo105 helicopter flight control is presented. The stability theorem for the entire fuzzy nonlinear system with optimal feedback gain is proposed [21]. A nonlinear dynamic Bo105 helicopter model is built first according to the aerodynamics analysis. A number of flight trajectory trim points are selected with the nonlinear flight model linearized around these trim points. The controllers for these linearized models are based on quadratic optimal technology. The TS fuzzy system is implemented as a gain-scheduler. Computer simulations are provided to verify the practicality of the TS fuzzy gain-scheduler acting on the Bo105 helicopter. The turning rate regulation and forward velocity tracking responses demonstrate the effectiveness of fuzzy gain scheduling.

## 14.2 The Nonlinear Dynamics of the Helicopter

This Bo105 helicopter has six degrees of freedom. Hence, the nonlinear dynamic equation for the helicopter is described as follow [15]:

$$\dot{u} = rv - qw + Xm^{-1} - g \sin \theta \quad (14.1)$$

$$\dot{v} = pw - ru + Ym^{-1} + g \sin \phi \cos \theta \quad (14.2)$$

$$\dot{w} = qu - pv + Zm^{-1} + g \cos \phi \cos \theta \quad (14.3)$$

$$\dot{\phi} = p + \tan \theta (q \sin \phi + r \cos \phi) \quad (14.4)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (14.5)$$

$$\dot{\varphi} = (q \sin \phi + r \cos \phi) \cos \theta^{-1} \quad (14.6)$$

where  $(u, v, w)$  and  $(p, q, r)$  are notations for the linear and angular velocity referred to the fuselage axes system. The symbol  $m$  is the mass of the helicopter, and  $(\theta, \phi, \varphi)$  are the Euler angles defined as the orientation of the body axes relative to the earth.

The equation of motion for the attitude of the helicopter is described as follow:

$$(I_{xx}I_{zz} - I_{xz}^2)\dot{p} = [(I_{yy} - I_{zz})I_{zz} - I_{xz}^2]q + (I_{xx} - I_{yy} + I_{zz})I_{xz}pq + I_{zz}L + I_{xz}N \quad (14.7)$$

$$I_{yy}\dot{q} = [(I_{zz} - I_{xx})pr - I_{xz}(p^2 - r^2) + M] \quad (14.8)$$

$$(I_{xx}I_{zz} - I_{xz}^2)\dot{r} = [(I_{xx} - I_{yy})I_{xx} + I_{xz}^2]pq - (I_{xx} - I_{yy} + I_{zz})I_{xz}rq + I_{xz}L + I_{xx}N \quad (14.9)$$

The variables ( $I_{xx}$ ,  $I_{zz}$ ,  $I_{xz}$ ) in Equations (14.7) ~ (14.9) consist of the inertial mass of the helicopter. The notations ( $X$ ,  $Y$ ,  $Z$ ) and ( $L$ ,  $M$ ,  $N$ ) are the forces and moments referred to the fuselage axes and can be described as the sum of the contribution from five sub-systems:

$$\begin{aligned} X &= X_R + X_T + X_F + X_{tp} + X_{fn} \\ Y &= Y_R + Y_T + Y_F + Y_{tp} + Y_{fn} \\ Z &= Z_R + Z_T + Z_F + Z_{tp} + Z_{fn} \\ L &= L_R + L_T + L_F + L_{tp} + L_{fn} \\ M &= M_R + M_T + M_F + M_{tp} + M_{fn} \\ N &= N_R + N_T + N_F + N_{tp} + N_{fn} \end{aligned} \quad (14.10)$$

The subscripts represent rotor ( $R$ ), tail rotor ( $T$ ), fuselage ( $F$ ), horizontal tail plane ( $tp$ ) and vertical fin ( $fn$ ). The aerodynamic forces and moments calculations for the five sub-systems refer to a coordinate system different from the fuselage coordinate system. It is necessary to calculate a transformation between the two coordinate systems [14]. In establishing the helicopter dynamics, the contribution of the downwash must be considered. The downwash component that is normal to the rotor disc is calculated in this study.

The Bo105 helicopter presents four pitch angles controlled by

$$U = [\theta_0 \quad \theta_{ls} \quad \theta_{lc} \quad \theta_{0T}]^T \quad (14.11)$$

where  $\theta_0$  is the collective pitch angle of the main rotor,  $\theta_{ls}$  is the longitudinal cyclic pitch angle of the main rotor,  $\theta_{lc}$  is the lateral cyclic pitch angle of the main rotor, and  $\theta_{0T}$  is the collective pitch angle of the tail rotor. This helicopter achieves maneuverability *via* manipulation of the four pitch angles in  $U$ . The rotor flapping dynamics are also included in the rotor sub-system derivation. The rotor forces and moments are not only a function of  $U$  but also a function of the rotor blade flap angle. Moreover, the flap angles can be obtained from the control input  $U$  and defined as

$$\beta = [\beta_0 \quad \beta_{ls} \quad \beta_{lc}]^T \quad (14.12)$$

where  $\beta_0$  indicates the coning angle,  $\beta_{ls}$  and  $\beta_{lc}$  indicate the longitudinal flapping angle and the lateral flapping angle, respectively. The notation at Equation (14.12)

is related to the multi-blade coordinate and stands for the behavior of all rotor blades. Similarly, the tail rotor flap angle must be considered in the tail rotor derivation. The fuselage forces and moments are functions of the fuselage incidence and sideslip angle. In the same way, the tail plane and fin forces and moments are functions of the respective incidence and sideslip angle. The modeling process is roughly discussed in this chapter and the helicopter modeling details can be found in [16].

The process for finding how to determinine the trimmed points will be discussed in the following. The trim and iteration method will be briefly stated. In the linearization process for a nonlinear system, the operating point must be determined; otherwise the linear model will exhibit residual input. The meaning of trim is to compute the equilibrium points of the dynamic equations for the helicopter. Let the state derivatives on the equilibrium points referring to the fuselage system be zero. Equations (14.1) ~ (14.9) can be rewritten as Equations (14.13) ~ (14.21):

$$rv - qw + Xm^{-1} - g \sin \theta = 0 \quad (14.13)$$

$$pw - ru + Ym^{-1} + g \sin \phi \cos \theta = 0 \quad (14.14)$$

$$qu - pv + Zm^{-1} + g \cos \phi \cos \theta = 0 \quad (14.15)$$

$$p + \dot{\phi} \sin \theta = 0 \quad (14.16)$$

$$q - \dot{\phi} \sin \theta \cos \theta = 0 \quad (14.17)$$

$$r - \dot{\phi} \cos \phi \cos \theta = 0 \quad (14.18)$$

$$(I_{yy} - I_{zz})qr + I_{xz}pq + L = 0 \quad (14.19)$$

$$(I_{zz} - I_{xx})pr + I_{xz}(r^2 - p^2) + M = 0 \quad (14.20)$$

$$(I_{xx} - I_{yy})pq + I_{xz}qr + N = 0 \quad (14.21)$$

The state variables ( $u, v, w, p, q, r, \theta, \phi, \dot{\phi}$ ) on the equilibrium points could be calculated *via* the above equations. Conversely, the control input  $U$  on the equilibrium points should be estimated for establishing linear models. However, there are only nine equations. Thus, an additional four trim parameters must be assumed for computing the control input. The four trim parameters are defined as flight speed  $V_{tr}$ , flight path angle  $\gamma_{tr}$ , turn rate  $\Omega_{tr}$ , and sideslip  $\beta_{tr}$  respectively.

The variables that are not defined in the process of calculation but will be utilized in advance must be guessed initially. The initially estimated variables will be continually iterated until the errors are smaller than a certain range. If the error is greater than the acceptable range, that variable will be set as follows:

$$\begin{aligned} \Theta_{new} &= \Theta_{initial} + \eta(\Theta - \Theta_{initial}) \\ \Theta_{initial} &= \Theta_{new} \end{aligned} \quad (14.22)$$

where  $\Theta_{initial}$  is the initial value which must be guessed first, and  $\Theta_{new}$  is the new value in the next iteration. The notation  $\Theta$  is the temporary value of the variable must be obtained in the iteration, and  $\Theta$  is obtained form Equations

(14.13)~(14.21). The notation  $\eta$  is the damping factor and is set as 0.5. The iteration process is continued until the error is smaller than the acceptable range:

$$|\Theta - \Theta_{initial}| \leq \varepsilon |\Theta| \quad (14.23)$$

where  $\varepsilon$  is the tolerance. If every condition is satisfied during the trim process and the errors are smaller than the acceptable range, the state variables and control input on the equilibrium points values can be determined. Linear models of the helicopter around the equilibrium points will be derived via Taylor's expansion after obtaining the equilibrium points.

### 14.3 Design of Linear Optimal Controller

Consider the nonlinear dynamic equation for the helicopter given by

$$\dot{x}(t) = F(x, U, t) \quad (14.24)$$

where the states  $x$  and controls  $U$  are

$$\begin{aligned} x &\equiv [u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \varphi]^T \\ U &\equiv [\theta_0 \quad \theta_{ls} \quad \theta_{lc} \quad \theta_{ot}]^T \end{aligned} \quad (14.25)$$

The small perturbation theory is applied to linearize the system. The helicopter state and control inputs can be described as a perturbation from the trim in the form

$$x = x_e + \delta x \quad (14.26)$$

where the subscript  $e$  is a trim notation. Assume that the external force ( $X, Y, Z$ ) and the moments ( $L, M, N$ ) can be represented as an analytical function of the state variables and control input. Using Taylor's expansion and neglecting the nonlinear terms, the linearized equation of motion around a general trim condition can then be written as

$$\dot{x}(t) = Ax(t) + BU(t) \quad (14.27)$$

where  $A$  means the helicopter system matrix and  $B$  represents the helicopter control matrix. The partial derivatives in  $B$  are called the control derivatives.

A number of linear models are needed for gain scheduling. Eight linear models are selected according to various forward flight conditions [0, 20, 40, 60, 80, 100, 120, 140] (knots/h). Equation (14.28) represents the helicopter hover system matrix  $A$  and the control matrix  $B$ . The eigenvalues of matrix  $A$  are composed of the roll subsidence, pitch subsidence, spiral subsidence, Dutch roll and phugoid mode. Table 14.1 lists the eigenvalues of the system matrix:

$$A = \begin{bmatrix} -0.02 & -0.02 & 0.01 & 0 & 0.9 & 0 & 0 & -9.8 \\ 0.02 & -0.035 & -0.002 & -0.8 & 0.0001 & 0 & 9.79 & 0.02 \\ 0.1 & 0 & -0.32 & 0 & 0 & 0 & 0.46 & -0.41 \\ 0.35 & -0.41 & 0 & -14 & 0.4 & 0.1 & 0 & 0 \\ 0.11 & 0.09 & -0.002 & -0.1 & -3.75 & 0 & 0 & 0 \\ 0.06 & -0.013 & 0.01 & -2.5 & 0.01 & -0.3 & 0 & 0 \\ 0 & 0 & 0 & 1 & -0.002 & 0.042 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.999 & 0.0467 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 3.8 & -8.5 & 3.3 & 0 \\ -0.5 & -3.4 & -9 & 5 \\ -92 & 0 & -0.5 & 0 \\ 5 & -69 & -172 & 6 \\ -5 & 45 & -17.5 & 0 \\ 20 & -12.2 & -30 & -15.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (14.28)$$

**Table 14.1.** The eigenvalues of system matrix

Roll subsidence	Pitch subsidence	Spiral subsidence	Dutch roll	Phugoid
-14.023	-3.8444	-0.3075 ± 0.03153j	-0.2138 ± 0.6087j	0.2410 ± 0.5233j

Consider the linear system shown in Equation (14.27), assuming that

$$U(t) = k(t)[r(t) - x(t)] \quad (14.29)$$

where  $k(t) \in R^{4 \times 8}$  is the state feedback matrix, and  $r$  is the reference command. The control objective is to make the states track the command closely and the difference between them approach zero. The error signal is defined as

$$e(t) = r(t) - x(t) \quad (14.30)$$

Differentiating both sides yields

$$\dot{e}(t) = \dot{r}(t) - \dot{x}(t) \quad (14.31)$$

Substituting Equation (14.27) and (14.29) into (14.31) produces

$$\dot{e}(t) - \dot{r}(t) = [A - Bk(t)]e(t) - Ar(t) \quad (14.32)$$

If  $\dot{r} = 0$  ( $r$  is a constant), the error signal will converge on zero if and only if  $eig(A - Bk(t)) < 0$ . In other words, we must find a state feedback controller that enables  $eig(A - Bk(t)) < 0$ . The LQ optimal control methodology is applied to design the linear controller. The purpose of optimal control is to find the weighting function of state  $Q \in R^{8 \times 8}$  and the weighting function of control variable  $R \in R^{4 \times 4}$  to make  $J$  minimal.  $J$  is the cost function and is defined as

$$J = \frac{1}{2} \int [x^T(t)Qx(t) + U^T(t)RU(t)]dt \quad (14.33)$$

where  $Q$  is a semi-positive definite matrix and  $R$  is a positive definite matrix. The control input  $U$  of the optimal controller is

$$\begin{aligned} U(t) &= R^{-1}B^T S(t)[r(t) - x(t)] \\ &= k(t)[r(t) - x(t)] \end{aligned} \quad (14.34)$$

where  $S$  is the solution of the Riccati equation

$$A^T S + SA - SBR^{-1}B^T S + Q = -\dot{S} \quad (14.35)$$

When time  $t$  in Equation (14.33) approaches infinity, Equation (14.35) can be rewritten as follows by setting  $\dot{S}$  equal to zero:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \quad (14.36)$$

Equation (14.36) is the so-called Riccati algebra equation. If the linear system shown in Equation (14.27) is controllable and time-invariant, there exists a unique and semi-positive definite constant matrix  $S$  [17]. Consequently, the  $k(t)$  in Equation (14.34) will be a constant “Kalman gain” and ensure that the closed-loop system is linearized on the trim conditions and exponentially stable and equivalent  $eig(A - Bk) < 0$  [18].

Because there are eight linear models based on various forward flight velocities, eight different weighting functions  $Q$  and  $R$  must be designed. We take the linear hover model for example; the chosen matrices  $Q$  and  $R$  are diag (30, 100, 800, 1, 50, 500, 2, 10) and diag (1000, 3000, 500, 100) respectively. The optimal controller is obtained after solving the Riccati algebra equation. The controller is tested to show the performance. Let the command signal  $u$  be a step signal and the simulation time set as 10 s. The initial conditions of the states are  $u = 0$ ,  $v = 0$ , and  $w = 0$ . The helicopter response is shown in Figure 14.1. Apparently, the forward velocity  $u$  follows the command signal. The other state will approach zero. The

turning rate regulation is presented next. The initial conditions of the three states are  $p = 0.1$ ,  $q = 0.5$ , and  $r = 0$ . Figure 14.2 shows the helicopter response after state feedback control. The turning rate converges on zero quickly. All of these response in this study are simulated by using MATLAB® Simulink® Toolbox.

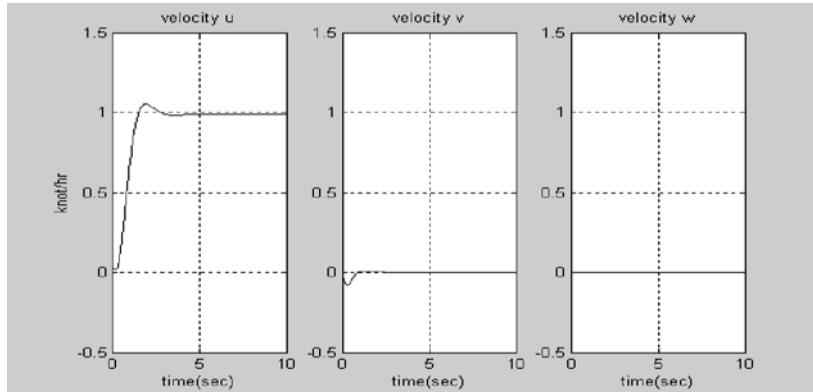


Figure 14.1. Helicopter response for step command

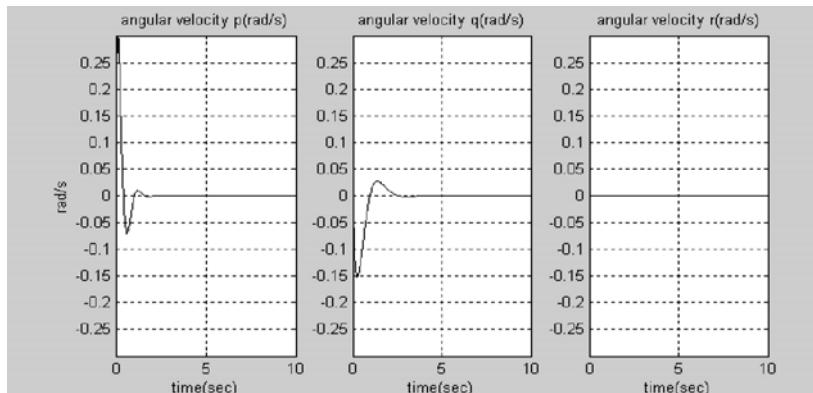


Figure 14.2. Helicopter response for turning rate regulation

## 14.4 Gain Scheduling Using Takagi-Sugeno Fuzzy Model

As mentioned in Section 14.2, there are eight linear models chosen in the flight envelope. The original nonlinear system can be replaced by a Takagi-Sugeno-type fuzzy system consisting of eight linear models. The designed TS fuzzy system is given as Model Rule  $R^i$

$$\begin{aligned} & \text{IF } x_1(t) \text{ is } M_{i1}(x_1) \text{ and } \dots \text{ and } x_j(t) \text{ is } M_{ij}(x_j) \quad i = 1, 2, \dots, 8. \\ & \text{THEN } \dot{x}(t) = A_i x(t) + B_i U(t) \end{aligned} \quad (14.37)$$

where  $R^i$  is the  $i$ th rule of the fuzzy system,  $x_j(t)$  is the state fed into the fuzzy system,  $M_{ij}(x_j)$  is the input membership function of  $x_j(t)$ , and  $A_i x(t) + B_i U(t)$  is the  $i$ th linear model. The TS fuzzy system controller is

$$\begin{aligned} & \text{IF } x_1(t) \text{ is } M_{i1}(x_1) \text{ and } \dots \text{ and } x_j(t) \text{ is } M_{ij}(x_j) \quad i = 1, 2, \dots, 8. \\ & \text{THEN } U(t) = k_i[r(t) - x(t)] \end{aligned} \quad (14.38)$$

where  $k_i$  is the  $i$ th optimal controller of the  $i$ th linear model.

The overall TS fuzzy system can be expressed by

$$\dot{x}(t) = \sum_{i=1}^8 \alpha_i [A_i x(t) + B_i U(t)] \quad (14.39)$$

where  $\alpha_i$  is the weight of the  $A_i x(t) + B_i U(t)$  on the whole fuzzy system and is defined by

$$\alpha_i = w_i(x) \left[ \sum_{i=1}^8 w_i(x) \right]^{-1} \quad (14.40)$$

where  $w_i(x) = \prod_{j=1}^n M_{ij}(x_j)$ . The overall fuzzy system controller  $K$  is given as

$$K = \sum_{i=1}^8 \alpha_i k_i \quad (14.41)$$

where  $\alpha_i$  is the weight of  $k_i$  on the whole controller  $K$ .

The stability details for the TS fuzzy system are carefully described in [19]. In this research, we focus only on the stability of a continuous and time-invariant fuzzy system with feedback controller. The properties are stated as follows:

**Theorem:** If every fuzzy subsystem in Equation (14.37) is time-invariant and controllable, the entire fuzzy nonlinear system then has optimal feedback gain:

$$\dot{x}(t) = \sum_{i=1}^8 \alpha_i (A_i - B_i k_i) x(t) \quad (14.42)$$

ensuring that closed loop system is exponentially global stable.

**Proof:** By the converse theorems from the Lyapunov stability theory, a nonlinear system is exponentially stable if and only if the linearized system around the equilibrium point of the nonlinear system is exponentially stable [20]. The stability of the entire nonlinear fuzzy system with optimal feedback gain on any equilibrium ( $X$ ) fed into the scheduler is the same as the linearized fuzzy system in the following:

$$\dot{x}(t) = \sum_{i=1}^8 \alpha_i(X)(A_i - B_i k_i)x(t) \quad (14.43)$$

As mentioned in Section 14.3, every closed-loop linear model is exponentially stable and implies that the spectrum of  $(A_i - B_i k_i)$  defined as  $\sigma(A_i - B_i k_i)$  is smaller than zero. Furthermore, via the “spectral mapping theorem” [18], we can conclude that for all  $X \in R^{n \times 1}$  and  $\alpha_i(X) \in [0, 1]$

$$\sigma[\alpha_i(X)(A_i - B_i k_i)] = \{\alpha_i(X)(\lambda_{i1}), \dots, \alpha_i(X)(\lambda_{i8})\} < 0 \quad (14.44)$$

From Equation (14.43), we can assume that the zero solution of  $\dot{x}(t) = \alpha_i(X)(A_i - B_i k_i)x(t)$  on  $t \mid t_0$  is exponentially stable for all  $X \in R^{n \times 1}$  and  $\alpha_i(X) \in [0, 1]$ . There exists fixed positive constants  $\gamma_i$  and  $m_i$  independent of  $t_0$  such that for all  $t_0 \in R^+$

$$\left\| e^{\alpha(X)(A-Bk)(t-t_0)} \right\| \leq m_i e^{-\gamma_i(t-t_0)} \quad \forall t \mid t_0 \quad (14.45)$$

The state transition matrix  $\phi(t, t_0)$  of Equation (14.42) is then

$$\begin{aligned} \|\phi(t, t_0)\| &= \left\| e^{\sum_{i=1}^8 \alpha_i(X)(A_i - B_i k_i)(t-t_0)} \right\| \\ &\leq \prod_{i=1}^8 \left\| e^{\alpha_i(X)(A_i - B_i k_i)(t-t_0)} \right\| \\ &\leq \prod_{i=1}^8 m_i e^{-\gamma_i(t-t_0)} = M e^{-\gamma(t-t_0)} \end{aligned} \quad (14.46)$$

where  $M \equiv \prod_{i=1}^8 m_i$  and  $\gamma \equiv \sum_{i=1}^8 \gamma_i$ . Therefore, the linearized fuzzy system (14.43) and entire fuzzy system (14.42) are exponentially global stable.

In the TS fuzzy gain scheduler design, three state variables are fed into the scheduler described as follows. The first channel is the forward velocity,  $x_1 = u$ . The second channel is the attitude angle,  $x_7 = \phi$ . The third channel is the pitch

angle,  $x_8 = \theta$ . The motivation for choosing the first specific state is that the obtained linear model is based on the forward flight speed, making this an important index to determine which linear model that state belongs to in the TS fuzzy system. Furthermore, the states  $\phi$  and  $\theta$  affect the helicopter flight speed and stability greatly and make the response unstable without confinement and control. The roll  $\phi$  and pitch angle  $\theta$  must be sent to the fuzzy gain scheduler for this reason. Because the linear models calculated the trim according to the forward flight speed of 0~140 knots/h, there are eight membership functions needed in the input 1 channel. The membership functions are shown in Figure 14.3. The fuzzy system in Equation (14.37) is significantly determined by input 1 and becomes

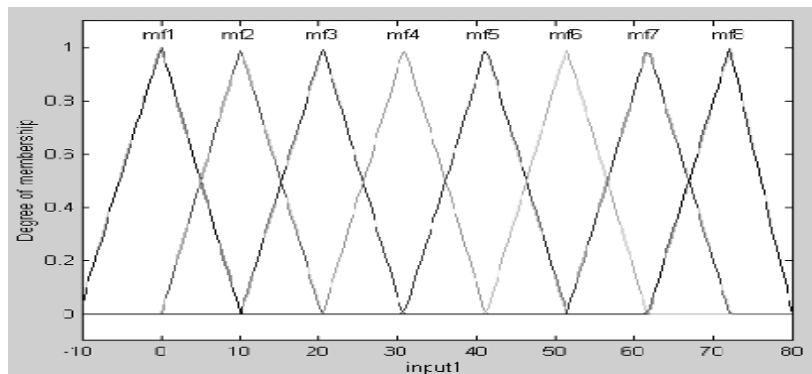
$$\begin{aligned} \text{IF } u \text{ is } M_i(u) \text{ and } \phi \text{ is } M_i(\phi) \text{ and } \theta \text{ is } M_i(\theta) \quad i = 1, 2, \dots, 8. \\ \text{THEN } \dot{x}(t) = A_i x(t) + B_i U(t) \end{aligned} \quad (14.47)$$

The controller for Equation (14.47) becomes

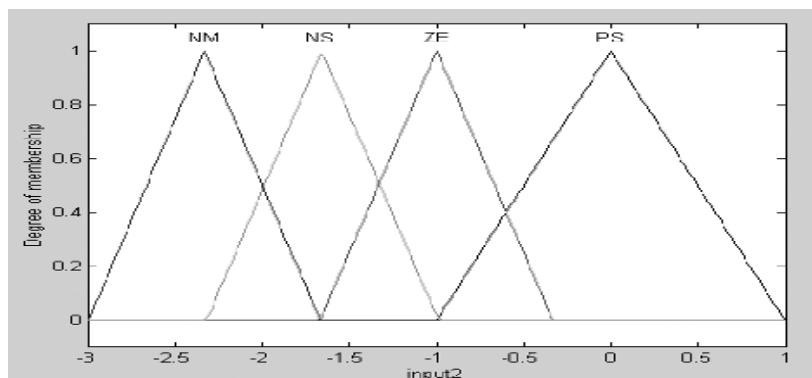
$$\begin{aligned} \text{IF } u \text{ is } M_i(u) \text{ and } \phi \text{ is } M_i(\phi) \text{ and } \theta \text{ is } M_i(\theta) \quad i = 1, 2, \dots, 8. \\ \text{THEN } U = k_i(r - x) \end{aligned} \quad (14.48)$$

The object of inputs 2 and 3 is to tune the proportion of  $k_i$  determined by input 1. The fuzzy gain scheduler output is a proportional constant for tuning the  $k_i$  on the entire  $K$  controller. In this study, the TS fuzzy gain scheduler is composed of eight Sugeno-type fuzzy controllers that tune the corresponding linear optimal controller  $k_i$ . Figures 14.4 and 14.5 show the membership functions for inputs 2 and 3, respectively. From the number of the membership functions for inputs 2 and 3, there will be at least 16 fuzzy rules with regard to every membership function for input 1. The fuzzy rules are established based on the designer's knowledge and experience.

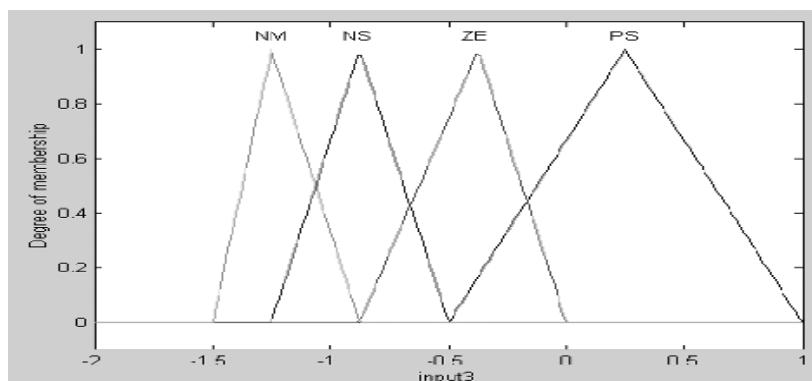
The designed TS gain scheduler is applied to the practical nonlinear dynamics of the Bo105 helicopter and various commands are considered to test the performance. First of all, the turning rate regulation is discussed and the response is shown in Figure 14.6 with initial condition  $p = 0.1$ ,  $q = 0.05$ , and  $r = 0$ . Then the discussion focuses on the forward flight control. Let the forward command velocity  $u$  be the step signal and the simulation time is set as 60 s. The other command is zero. The initial conditions of the states  $u$ ,  $v$ , and  $w$  are  $[0, 0, 0]$ . Figure 14.7 shows the helicopter flight response using fuzzy gain scheduling. The forward velocity  $u$  tracks the command signal over the flight envelope. The minor steady state errors occur due to the residual forces and moments in the trim analysis for  $v$  and  $w$ . Furthermore, let the forward velocity command  $u$  be the ramp signal and the simulation time be set at 200 s. The initial conditions of the states  $u$ ,  $v$ , and  $w$  are  $[0, 0, 0]$ . Figure 14.8 shows that the forward velocity  $u$  tracks the command signal  $u$  well. Hence, the TS fuzzy gain scheduler can be utilized to control the nonlinear dynamics of helicopter Bo105.



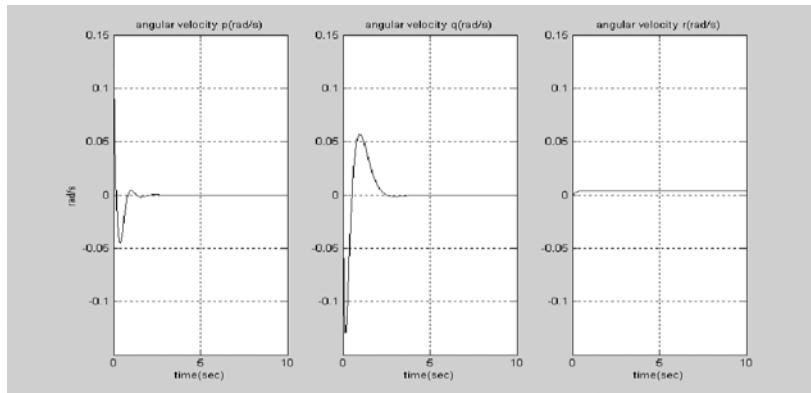
**Figure 14.3.** Membership functions of input 1



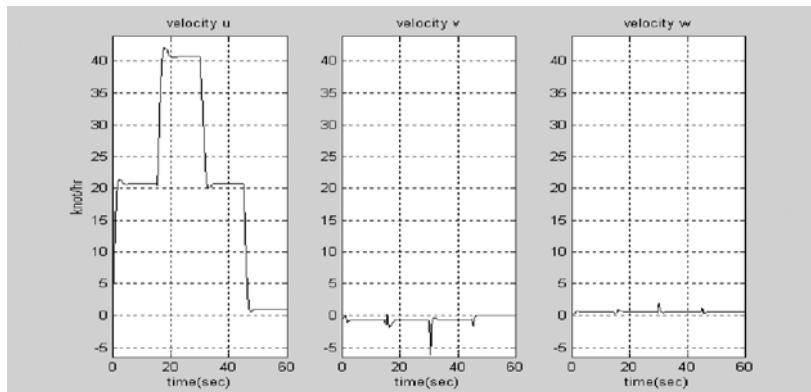
**Figure 14.4.** Membership functions of input 2



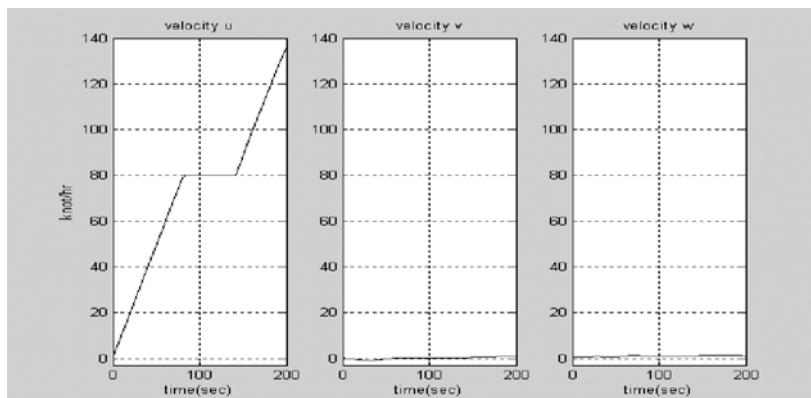
**Figure 14.5.** Membership functions of input 3



**Figure 14.6.** The regulation response of the turning rate using gain-scheduler



**Figure 14.7.** The step response of the forward velocity using gain-scheduler



**Figure 14.8.** The ramp response of the forward velocity using gain-scheduler

## 14.5 Conclusions

The Takagi-Sugeno fuzzy gain-scheduler is designed for nonlinear optimal flight control of the Bo105 helicopter. The stability theorem for the entire fuzzy system with optimal feedback controllers has been validated. After the aerodynamics analysis, the nonlinear helicopter model is built first. The nonlinear dynamics of the helicopter are linearized around trim points. Various optimal controllers are designed based on the linearized model. The TS fuzzy system is implemented as a gain-scheduler. The first input for the fuzzy gain-scheduler determines which linearized model will be activated. The second and third inputs are implemented to tune the local linear controller proportion in the global gain-scheduling system. Computer simulation demonstrates the effectiveness of the TS fuzzy gain-scheduler. The regulation response of the turning rate exhibits excellent performance of fuzzy gain scheduling. In forward flight studies, the helicopter response can track both the step and ramp command regardless of the amount of forward velocity.

## Acknowledgement

This research is sponsored in part by means of the National Science Council of the Republic of China, under Grant NSC-94-2213-E-197-004-.

## References

- [1] D. McLean, *Automatic Flight Control System*, Prentice Hall, 1990, pp 451-489.
- [2] F. L. Lewis and V. L Syrmos, *Optimal Control*, Wiley., 1995, pp 359-376.
- [3] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*, Prentice Hall, 1990, pp 101-125.
- [4] M. Margaliot and G. Langholz, "Hyperbolic Optimal Control and Fuzzy Control," *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, Vol. 29, No. 1, January 1999, pp 1-10.
- [5] B. Kadmire, P. Bergsten, and D. Driankov, "Autonomous Helicopter Control Using Fuzzy Gain Scheduling," *Proceedings of the 2001 IEEE International Conference Robotics & Automation Seoul*, Korea, May 21-26, 2001, pp 2981-2985.
- [6] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control, and Information*, Prentice-Hall, New Jersey; 1999, pp 4-6.
- [7] K. Tanaka and M. Sugeno, "Stability Analysis and Design of Fuzzy Control Systems," *Fuzzy Sets and System*, Vol. 45, No. 2, 1992, pp 135-156.
- [8] L. Fortuna, G. Rizzotto, M. Lavorghna, G. Nunnari, M. G. Xibilia, and R. Caponetto, *Soft Computing*, Springer, 2000, pp 5-21.
- [9] B. Kadmire and D. Driankov, "Autonomous Helicopter Control Using Linguistic and Model-Based Fuzzy Control," *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, Mexico City, September 5-7, 2001, pp 348-352.
- [10] D. Driankov, R. Palm, and U. Rehfuss, "A Takagi-Sugeno Fuzzy Gain-Scheduler," *Proceedings of IEEE Conference Fuzzy Systems*, New Orleans, Florida, USA, 1996, pp 1053-1059.

- [11] C. Calvalcante, J. Cardoso, J. J. G. Ramos, and O. R. Neves, "Design and Tuning of a Helicopter Fuzzy Controller," *IEEE Int. Conference on Fuzzy Systems*, 1995, pp 1549-1554.
- [12] D. McLean and H. Matsuda, "Helicopter Station-Keeping: Comparing LQR, Fuzzy Logic and Neural-Net Controllers," *Engineering Applications of Artificial Intelligence*, Vol. 11, 1998, pp 411-418.
- [13] C. Phillips, C. Karr, and G. Walker, "Helicopter flight control: with fuzzy logic and genetic algorithms," *Engineering Applications of Artificial Intelligence*, Vol. 9, 1996, pp 175-184.
- [14] K. Sprague, V. Gavrilets, D. Dugail, B. Matter, and E. Feron, "Design and Applications of an Avionics System for a Miniature Acrobatic Helicopter," *DASC, the 20<sup>th</sup> Conference*, 2001, pp 3.C.5/1-3.C.5/10
- [15] G. D. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*, AIAA, 1996, pp 87-276.
- [16] D. W. Chiou, C. D. Yang, and R. F. Liu, "Establishing Helicopter Nonlinear Flight Simulation Environment," *Transactions of the Aeronautical Society of the Republic of China*, Vol. 32, No. 3, July 2000, pp 203-221.
- [17] E. R. Pinch, *Optimal Control And The Calculus of Variations*, Oxford Science Publication; 1993, pp 163-168.
- [18] F. M. Callier and C. A. Desor, *Linear System Theory*, New York: Springer-Verlag, 1991, pp 330-346.
- [19] S. J. Wu and C. T. Lin, "Optimal Fuzzy Controller Design: Local Concept Approach," *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 2, April 2000, pp 171-185.
- [20] M. Vidyasagar, *Nonlinear System Analysis*, Prentice Hall, 1993, pp 135-264.
- [21] Gwo-Ruey Yu, "Nonlinear Optimal Control of Helicopter Using Fuzzy Gain Scheduling," *IEEE Int. Conference on Systems, Man and Cybernetics*, 2005, pp 3313-3318.

## **Fuzzy Logic for Flight Control II: Fuzzy Logic Approach to Path Tracking and Obstacles Avoidance of UAVs**

Zhao Sun, Tao Dong, Xiaohong Liao, Ran Zhang and David Y. Song

### **15.1 Introduction**

Unmanned Aerial Vehicles (UAVs) have attracted increasing attention in military and civilian applications, such as mapping, patrolling, search and rescue, and reconnaissance. These tasks may be repetitive, dangerous, or both, making UAVs the ideal tools [1-9]. In these types of applications, the following issues are fundamental:

- 1) Path tracking
- 2) Obstacle avoidance (including collision avoidance with other aircrafts)

The problem considered here is how to make a UAV track a given flight trajectory while at the same time avoid unexpected obstacle(s)?

If the information about the existence of the obstacle is known in advance, then the problem can be readily solved by carefully pre-planning the desired flight path for the vehicle [3-5]

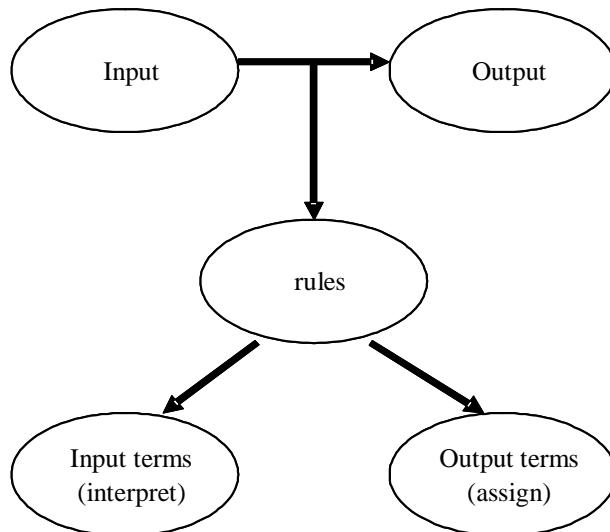
Recently, some methods of UAV/mobile vehicle navigation for obstacle avoidance have been reported [6-8]. Note that in [6] only a single point mass obstacle is considered. In [7, 8], no moving obstacle is addressed.

This chapter addresses the problem of how to make a UAV track a given flight trajectory while at the same time avoid unexpected obstacle(s). If the information about the existence of the obstacle is known in advance, then the problem can be readily solved by carefully pre-planning the desired flight path for the vehicle. However, most of the real situation is that obstacles might appear unexpectedly, and as such the pre-designed path might become “misleading” to the vehicle if no correcting action is taken. Furthermore, obstacles may be of any shape (not necessarily a point mass) and may be either inert or hostile (including seeking a collision). How to avoid unexpected, irregular, even moving obstacles while maintaining close path tracking of UAV is a challenging problem.

In this chapter, we explore a fuzzy logic-based approach to path tracking and obstacle avoiding. We consider the case that obstacles are either still or moving and appear along the pre-determined flight path unexpectedly. By using suitable sensors we identify the relative distance between the vehicle and the obstacle and

make timely adjustment to the pre-planned flight path. Fuzzy logic control algorithms are developed to achieve close path tracking while avoiding obstacles.

An interesting feature of fuzzy control system is that it is easy to deal with uncertain situations by representing the input and output relation in the “*if-then*” manner and constructing a knowledge base. The relation between inputs and outputs in fuzzy rules is shown in Figure 15.1.

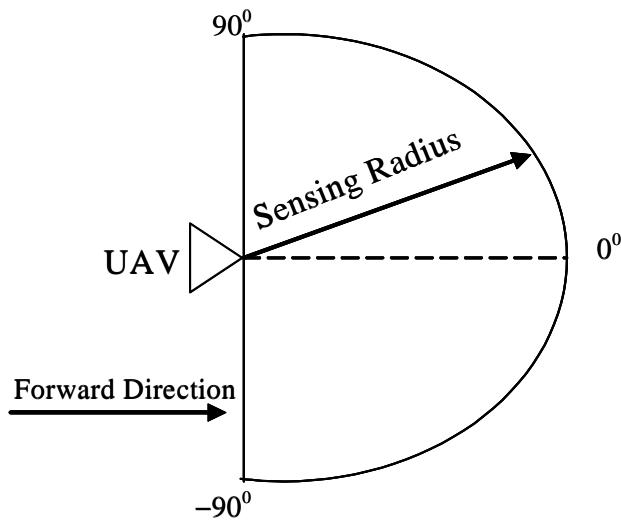


**Figure 15.1.** Fuzzy rules

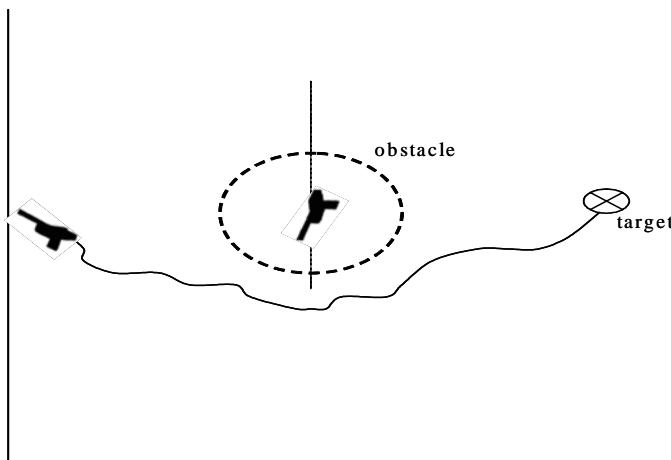
Due to the ability to handle unknown conditions, fuzzy logic system is an ideal tool to address the obstacle avoidance problem. In this chapter we combine model-based control with fuzzy logic control for UAV navigation in the presence of stationary or moving obstacles. This method provides a function for avoiding stationary and moving obstacles by sensing the distance and the angle between the UAV and the obstacles. Once an obstacle has been detected, the previous flight path is corrected to ensure safe flight while minimizing the deviation of UAV from the pre-planned path. Simulation studies on multiple obstacles with various shapes are conducted and the effectiveness of the proposed method is verified.

## 15.2 Strategy for Path Tracking and Obstacle Avoidance

We consider the problem of UAV trajectory tracking without detail information about the flight environment – only the position of the UAV is available from a GPS (Global Position System) receiver. We assume that the obstacles can be detected by a UAV obstacle sensor. The sensor range is shown in Figure 15.2.



(a)



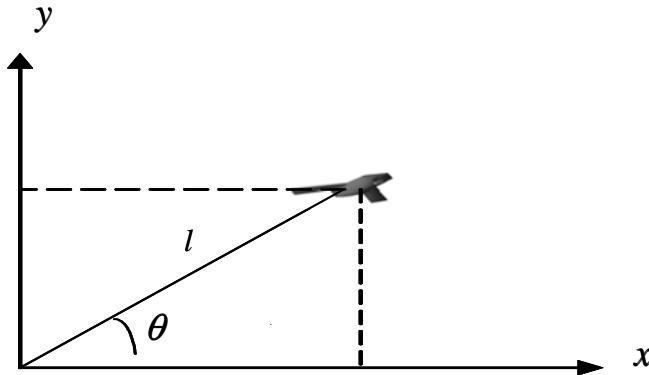
(b)

**Figure 15.2.** (a) Sensor range of UAV. (b) Example of obstacle avoidance

We are interested in maintaining close tracking of heading speed and heading angle of UAV, which are governed via

$$\begin{aligned}\dot{v} &= f_v(\cdot) + g_v(\cdot)v_c \\ \ddot{\theta} &= f_\theta(\cdot) + g_\theta(\cdot)\theta_c\end{aligned}\tag{15.1}$$

where  $v_c$  and  $\theta_c$  are the commanded velocity and heading angle to the autopilots.  $f_v(\cdot)$  and  $f_\theta(\cdot)$  are the nonlinear functions of the system,  $g_v(\cdot)$  and  $g_\theta(\cdot)$  represent the control gains.



**Figure 15.3.** Lateral-longitudinal coordinates

With the lateral and longitudinal coordinates as shown Figure 15.3, we can establish

$$\begin{cases} x = l \cos \theta \\ y = l \sin \theta \end{cases}\tag{15.2}$$

where  $\dot{l} = v$ , in which  $l$  is the distance between the UAV and the departure point, and  $x$  and  $y$  are the lateral and longitudinal positions of UAV with respect to the departure point. Clearly we have

$$\begin{cases} \dot{x} = v \cos \theta - (l \sin \theta) \dot{\theta} \\ \dot{y} = v \sin \theta + (l \cos \theta) \dot{\theta} \end{cases}\tag{15.3}$$

By defining  $X = (x \ y)^T$ ,  $U = (v_c \ \theta_c)^T$ , and

$$J = \begin{pmatrix} \cos \theta & -l \sin \theta \\ \sin \theta & l \cos \theta \end{pmatrix},$$

we get

$$\dot{X} = J \begin{pmatrix} v \\ \dot{\theta} \end{pmatrix}, \quad (15.4)$$

$$\ddot{X} = J \begin{pmatrix} v \\ \dot{\theta} \end{pmatrix} + J \begin{pmatrix} \dot{v} \\ \ddot{\theta} \end{pmatrix} \quad (15.5)$$

where

$$J = \begin{pmatrix} -\sin \theta \dot{\theta} & -v \sin \theta - l \cos \theta \dot{\theta} \\ \cos \theta \dot{\theta} & v \cos \theta - l \sin \theta \dot{\theta} \end{pmatrix}.$$

Combining Equations (15.1) and (15.5) leads to

$$\ddot{X} = F + GU, \quad (15.6)$$

where

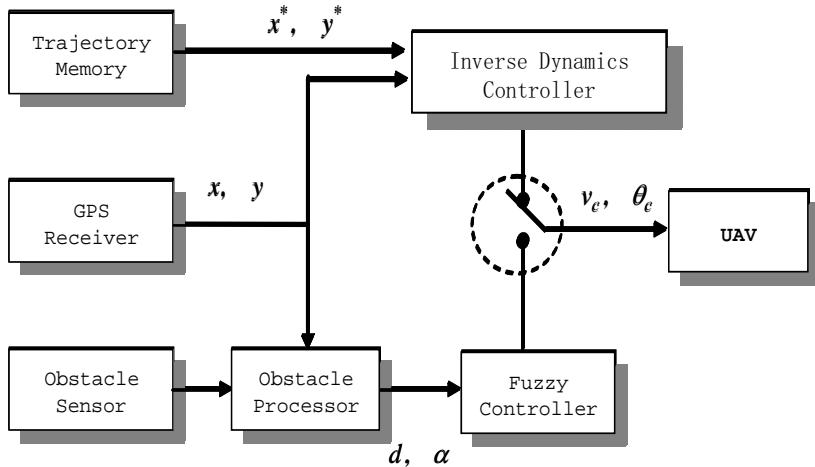
$$F = J \begin{pmatrix} v \\ \dot{\theta} \end{pmatrix} + J \begin{pmatrix} f_v(\cdot) & 0 \\ 0 & f_\theta(\cdot) \end{pmatrix}$$

and

$$G = J \begin{pmatrix} g_v & 0 \\ 0 & g_\theta \end{pmatrix}$$

The control objective is to design the control input  $U$  such that the actual flight path tracks the desired trajectory, *i.e.*,  $X \rightarrow X^*$ , where  $X^* = \begin{pmatrix} x^* & y^* \end{pmatrix}^T$  while avoiding unexpected obstacles.

The control scheme consists of two major parts. The first part is derived using the inverse dynamics method, which functions as the steering policy for regular UAV flight. The second part is a fuzzy logic based scheme which is synthesized to achieve obstacle avoidance, as shown in Figure 15.4.



**Figure 15.4.** Block diagram of UAV autopilot system

Mathematically, the autopilot signal is computed by

$$U = \eta U_I(.) + (1 - \eta) U_F(.) \quad (15.7)$$

where  $\eta = [0, 1]$  is a switching variable,  $U_I(.)$  is the inverse dynamics control generated via

$$U = G^{-1} \{ -k[(\dot{X} - \dot{X}^*) + \beta(X - X^*)] + \ddot{X}^* - \beta(\dot{X} - \dot{X}^*) - F \}, \quad k \neq 0 \quad (15.8)$$

and  $U_F(.)$  is the fuzzy logic controller (FLC), which is activated when the obstacle sensor detects any obstacle. The FLC will generate UAV velocity and heading angle adjustment corresponding to different situations. The fuzzy control rules are given in the following form:

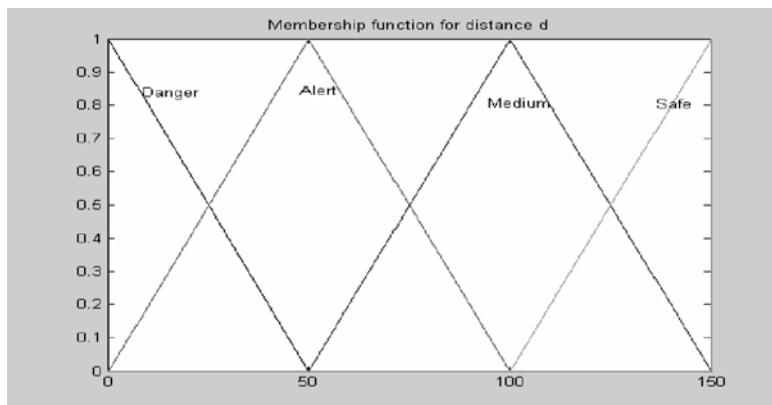
Rule  $i$ : if  $d$  is  $F_d^i$  and  $\alpha$  is  $F_\alpha^i$ , then  $u$  is  $\rho_i$ , where  $i = 1, 2, \dots, n$  are the singleton control actions and  $F_d^i$  and  $F_\alpha^i$  are the labels of the fuzzy sets.

The defuzzification of the controller outputs is accomplished by the method of center of gravity:

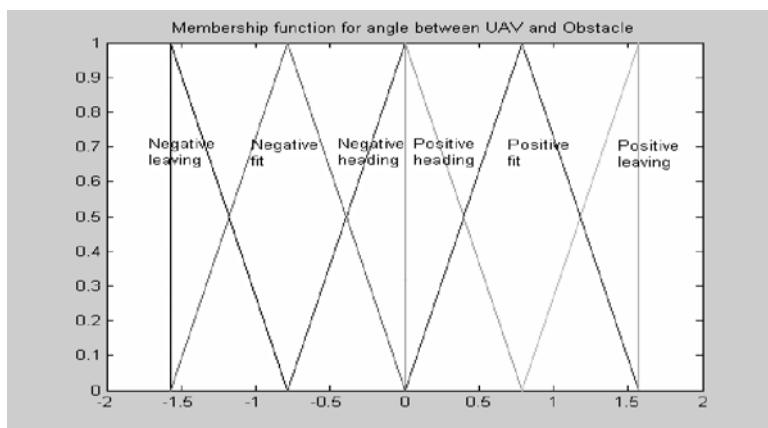
$$U_{flc}[d, \alpha, \rho_i] = \frac{\sum_{i=1}^n \phi_i \times \rho_i}{\sum_{i=1}^n \phi_i} \quad (15.9)$$

where  $\phi_i$  is the firing weights of the  $i$ th rule.  $d$  is the distance from obstacle and  $\alpha$  is the angle with respect to the obstacle. The fuzzy rules can be constructed by two parameters, their distance  $d$  from obstacles and their angle  $\alpha$  from obstacle. The distance  $d$  is in feet from the center of UAV. A distance of approximately 150 feet indicates that UAV is safe. The angle  $\alpha$  of approximately  $\pm\pi/4$  is a safe angle for UAV avoiding obstacles without large overshoot. With the model-based controller and fuzzy logic controller, the UAV system block diagram is shown in Figure 15.4.

The rule base is constructed based on human experience. Here in this case triangular membership functions have been chosen. Changing the membership function would change the performance of the system.



**Figure 15.5.** Input variable distance  $d$



**Figure 15.6.** Input variable angle  $\alpha$

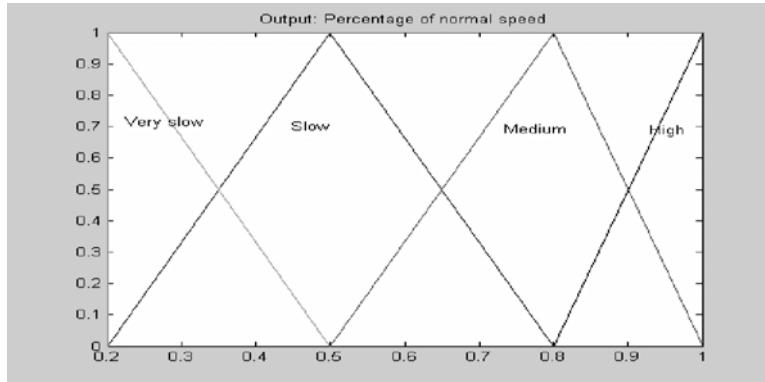


Figure 15.7. Output variable velocity  $v_C$

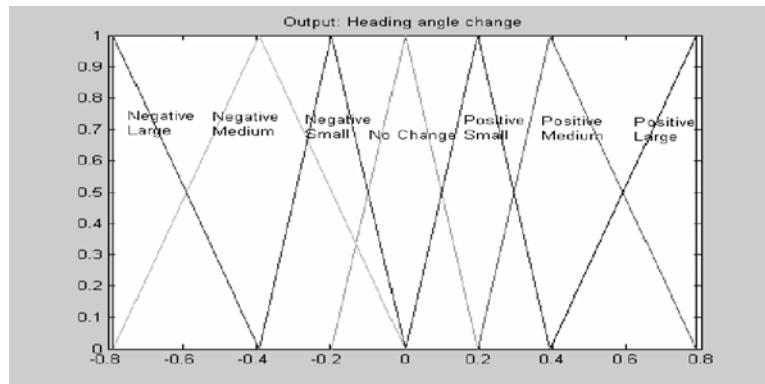


Figure 15.8. Output variable heading Angle  $\theta_C$

The inputs variables “ $d$ ” in this system can be divided into a range of “states”, such as: “Danger”, “Alert”, “Medium” and “Safe”, “ $\alpha$ ” divided into “Negative Leaving”, “Negative Fit”, “Negative Heading”, “Positive Heading”, “Positive Fit” and “Positive Leaving”. The outputs velocity change can be divided into “Very Slow”, “Slow”, “Medium”, and “High” and the change of the heading angle can be divided into “Negative Large”, “Negative Medium”, “Negative Small”, “No Change”, “Positive Small”, “Positive Medium” and “Positive Large”. We define the inputs and outputs states using “membership functions” shown as Figures 15.5, 15.6, 15.7 and 15.8.

Using the knowledge base we develop the fuzzy control rules “*if ... and ... then*” as shown in Tables 15.1 and 15.2 for the fuzzy outputs  $v_C$  and  $\theta_C$ , correspondingly. Those rules are integrated with the inverse dynamics control as described earlier to achieve path tracking and at the same time avoiding unexpected obstacles.

**Table 15.1.** Output velocity change

In 2 In 1	PL	PF	PH	NH	NF	NL
S	H	H	M	M	H	H
M	M	M	S	S	M	M
A	S	V	V	V	V	S
D	V	V	V	V	V	V

**Table 15.2.** Output heading angle change

In 2 In 1	PL	PF	PH	NH	NF	NL
S	PS	O	NS	PS	O	NS
M	PS	O	NM	PM	O	NS
A	O	NS	NL	PL	PS	O
D	NS	NM	NL	PL	PM	PS

### 15.3 Simulation Results

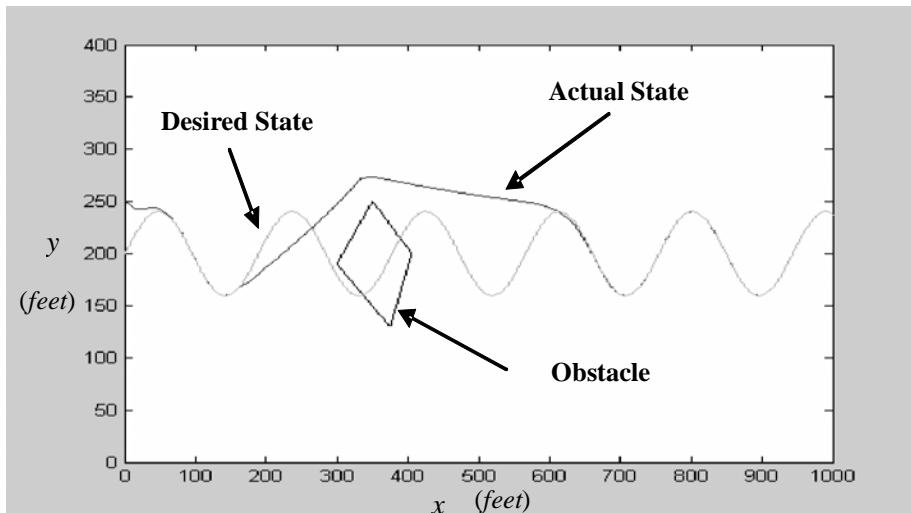
To test the effectiveness of the developed strategy, we conduct simulation on path tracking in the presence of obstacles with various shapes. Figure 15.9 is the path tracking process involving a quadrilateral obstacle along the path. As can be seen, if the UAV is steered to track the pre-determined (desired) trajectory, it would collide with the obstacle. By applying the proposed strategy, the actual path is adjusted so that the obstacle avoidance is achieved. The simulation also tested other types (shapes) of unexpected obstacles along the flight path (Figures 15.10 and 15.11).

In addition, we tested the multiple obstacles avoidance, as shown in Figures 15.12 and 15.13. It is observed that the control strategy works fairly well for single obstacle and multiple obstacles appearing unexpectedly.

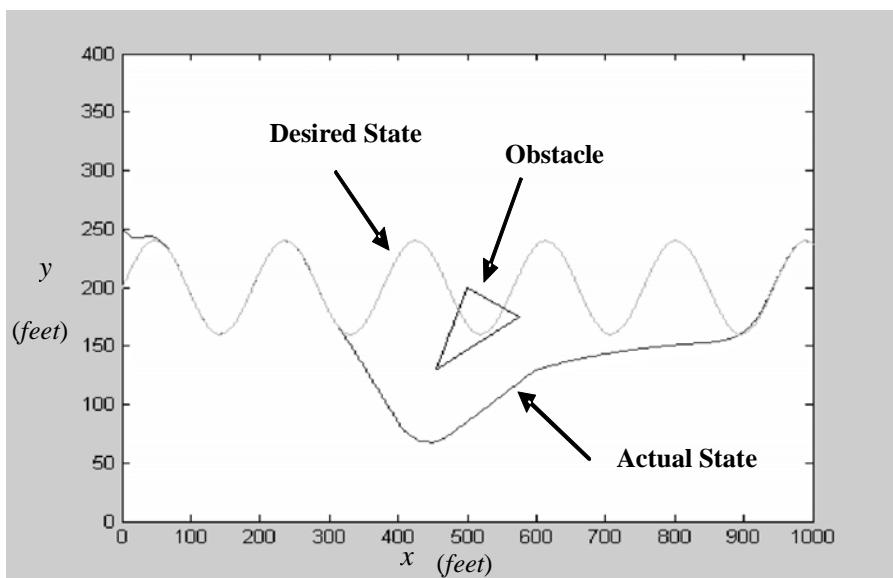
### 15.4 Conclusions

This chapter presented a fuzzy logic approach to path tracking and multi-obstacle avoidance. A two-layered FLC was used to make the UAV track its path while avoiding unexpected obstacles. The sensor inputs were fed in as random values and the desired path (commands) values for the UAVs were obtained, which were used

to guide and steer the vehicle so that its actual flying path is corrected/adjusted to bypass the obstacle. Simulation studies were conducted and verified the effectiveness of the proposed method.



**Figure 15.9.** Path tracking with one obstacle – shape 1



**Figure 15.10.** Path tracking with one obstacle- shape 2

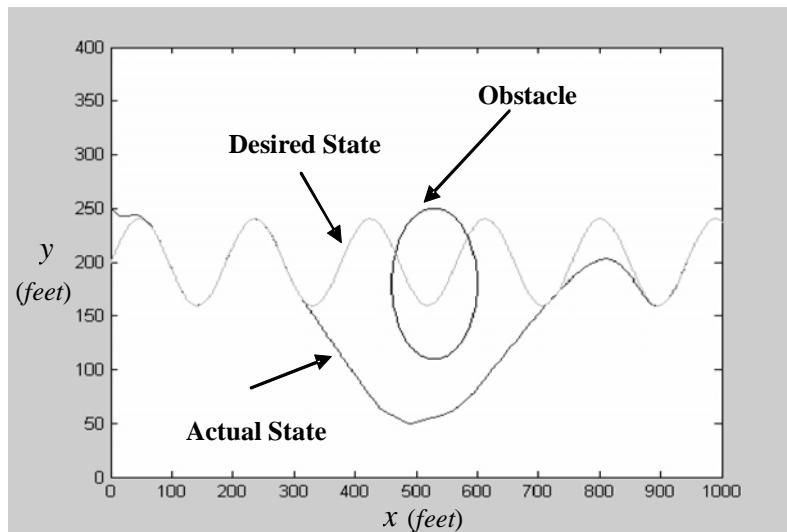


Figure 15.11. Path tracking with one obstacle- shape 3

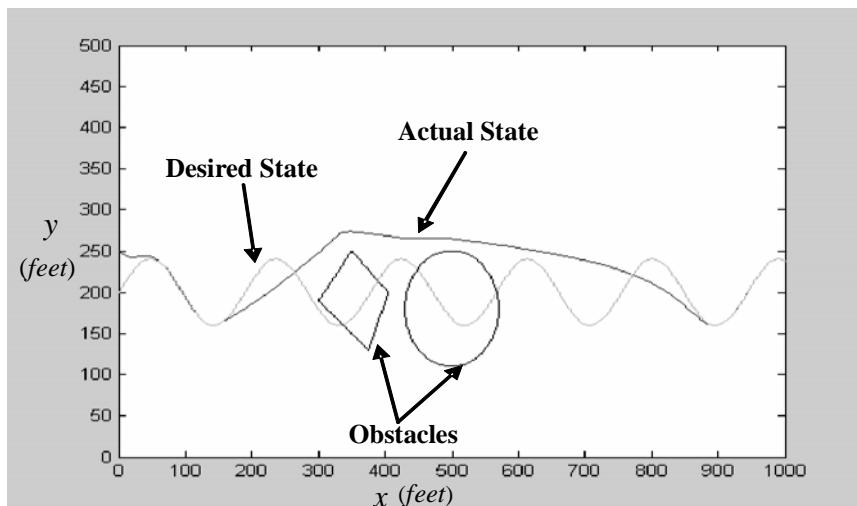
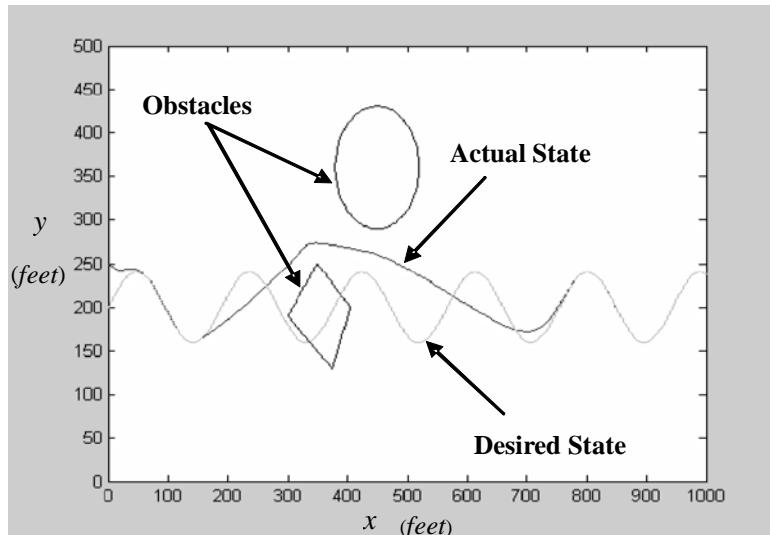


Figure 15.12. Path tracking with two obstacles along the same line



**Figure 15.13.** Path tracking with two obstacles across the flight path

## Acknowledgement

The work was supported in part by ONR under the contract N0014-03-1-0462 and NSF under the contract HRD-0450203.

## References

- [1] D.A. Schoenwald. UAVs: in Space, Air Water, and on the Ground. *IEEE Control System Magazine*, Vol. 20, No. 6, December 2000, pp 15-18.
- [2] Timothy W. McLain, Phillip R. Chandler, Steven Rasmussen and Meir Pachter Cooperative Control of UAV, *Rendezvous Proceedings of the American Control Conference*, Arlington, VA June 25-27, 2001.
- [3] D. Manicini, M. Brescia, E. Cascone, and P. Schipani, A Variable Structure Control Law for Telescopes Pointing and Tracking in Acquisition, Tracking, and pointing XI, ed. M. K. Masten, L.A. Stockum, Proc. SPIE 3086, 1997, pp 72-84.
- [4] C. Scali, G. Nardi, A. Landi, and A. Balestrino, Performance of Variable Structure PI controllers in Presence of Uncertainty and Saturation nonlinearities, XII IFAC World Congress, Sidney, Australia, Vol. 8, 1993. pp 527-532.
- [5] W. Gao and J. C. Hung, Variable Structure Control of Nonlinear System: A New Approach, *IEEE Trans. Industrial Electronics*, Vol. 40, Feb.1993, pp 45-55.

- [6] Yoichiyo Maeda and Morikazu Takegami, Collision Avoidance Control among Moving Obstacles for a Mobile Robot on the Fuzzy Reasoning, *Journal of the Robotic Society of Japan*, Vol. 6, No. 6, pp 518-522, Dec, 1988.
- [7] Tomoyoshi Takeuchi, An Autonomous Fuzzy Mobile Robot, *Journal of the Robotic Society of Japan*, Vol. 6, No. 6, pp 536-541, Dec. 1988.
- [8] Toshiaki Murofushi and Michio Sugeno, ‘Fuzzy Control of Model Car”, *Journal of the Robotic Society of Japan*, Vol. 6, No. 6, pp 536-541. Dec. 1988.
- [9] L. E. Buzogany, M. Pachter, J. J. D’Azzo: Automated Control of Aircraft in Formation Flight; *AIAA Guidance, Navigation, and Control Conference*, Part. 3, pp 1349-1370, 1993.

## 16

---

# Close Formation Flight Control of Multi-UAVs via Fuzzy Logic Technique

Yao Li, Bin Li, Zhao Sun, Liguo Weng, Ran Zhang, and David Y. Song

## 16.1 Introduction

In recent years, the design and implementation of control algorithms for unmanned air vehicles have been the object of quite a number of studies. There exist several approaches related to navigation and control of UAVs. Briefly, the corresponding research includes nonlinear adaptive control, neural networks, fuzzy logic, neuro-fuzzy control, genetic algorithms, and adaptive neural networks [1-8]. In this chapter we develop a new control strategy by integrating fuzzy logic control with pursuit guidance for anonymous UAV following leader-Wingman formation flight control.

Most of the results are based on flight models that linearize the nonlinear flight dynamics and/or ignore the effect of vortex. The presence of the vortex causes large disturbances to the trailing airplane, particularly in the roll axis, and a great deal of pilot workload is required to trim the airplane. To reduce demands on the pilot, significant research has been done in the development of automatic guidance and control laws for close formation flight as in Figure 16.1.



**Figure 16.1.** Close formation of manned flight vehicles

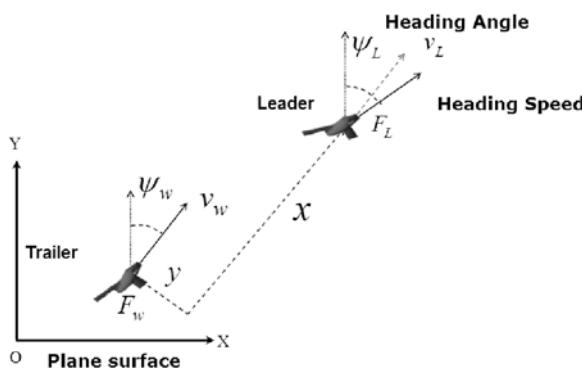
These controllers typically assume that timely and accurate measurements are exchanged between the aircraft. Some of the hurdles to close formation flight in an operational environment are the cost and logistics required to provide a

communication link between two aircraft that may be of different types and operated by different organizations.

In this chapter, two-module fuzzy logic control algorithms capable of maintaining desirable separation of wingman with the leading UAV are developed. The proposed method of trajectory tracking doesn't require the communication link, and compared with most existing methods, this formation control scheme is simpler in structure and easier to expand for implementation for other dynamic models.

## 16.2 Flight Model

The formation geometry is determined by the relative position between the Leader and Wingman as shown in Figure 16.2. The formation control objective is to steer the Wingman (follower) to maintain a certain separation distance in longitudinal, lateral and vertical directions. By properly defining the body frame and inertia frame, the three equations given below describing dynamic behavior of wingman aircraft can be established [1].



**Figure 16.2.** Formation flight in inertial frame

The definitions and the subscripts of the variables used in this chapter are listed in Tables 16.1 and 16.2 respectively.

**Table 16.1.** Nomenclature of the variables

Parameter	Variable
X Coordinate of aircraft	$x$
Y Coordinate of aircraft	$y$
Altitude	$h$
Velocity	$V$
Heading angle	$\psi$
Autopilot time constant	$\tau$

**Table 16.2.** Subscripts of the variables

Parameter	Subscript
Desired value	$d$
Separation / difference	$e$
Control signal	$c$
Leader aircraft	$l$
Wingman aircraft	$w$

It is assumed that each wingman aircraft could access the leader's complete flight condition at any time, and they are equipped with the same autopilot system which could execute the same command simultaneously:

$$\dot{V}_w = \frac{1}{\tau_V} V_{wc} - \frac{1}{\tau_V} V_w + \delta_1(.) \quad (16.1)$$

$$\dot{\psi}_w = \frac{1}{\tau_\psi} \psi_w + \frac{1}{\tau_\psi} \psi_{wc} + \delta_2(.) \quad (16.2)$$

$$\ddot{h}_w = -\left(\frac{1}{\tau_{ha}} + \frac{1}{\tau_{hb}}\right) \dot{h}_w - \frac{1}{\tau_{ha} \tau_{hb}} h_w + \frac{1}{\tau_{ha} \tau_{hb}} h_{wc} + \delta_3(.) \quad (16.3)$$

where  $V_w$ ,  $\psi_w$  and  $h_w$  are the wingman velocity, heading angle and altitude respectively. The uncertain dynamics in the lateral/longitudinal and altitude directions are modeled by using  $\delta_i(.)$  ( $i=1, 2, 3$ ). The detailed definitions of other variables can be found in [1]. In the planar reference coordination where leader is the origin, the relative velocity and separation distance of wingman aircraft can be decoupled as the following dynamic equations:

$$\dot{x} = -V_w + V_l \cos(\psi_l - \psi_w) + \dot{\psi}_w y \quad (16.4)$$

$$\dot{y} = V_l \sin(\psi_l - \psi_w) + \dot{\psi}_w x \quad (16.5)$$

Substituting Equations (16.1) and (16.2) into Equations (16.4) and (16.5), give the wingman's complete kinematical equations as follows:

$$\dot{x} = \left[ -\frac{y}{\tau_\psi} \psi_w + \frac{y}{\tau_\psi} \psi_{wc} + V_l \cos(\psi_l - \psi_w) \right] - V_w \quad (16.6)$$

$$\dot{y} = \frac{x}{\tau_\psi} \psi_w + V_l \sin(\psi_l - \psi_w) - \frac{x}{\tau_\psi} \psi_{wc} \quad (16.7)$$

where  $V_{wc}$  and  $\psi_{wc}$  are the control laws of wingman's velocity tracking and heading angle tracking.

### 16.3 Fuzzy Logic Control

One of the important tasks in close formation is to separate the wingman away from the leading UAV with certain distance in lateral ( $x$ -axis), longitudinal ( $y$ -axis) and vertical ( $h$ -axis) directions to prevent possible collision. To address this issue, a relative frame as shown in Figure 16.2 is considered. The relative coordinates are defined by

$$x_r = x_l - x_w,$$

$$y_r = y_l - y_w,$$

$$h_r = h_l - h_w.$$

The formation control problem can be stated as follows. Design control algorithms to maintain the wingman's relative position ( $x_r, y_r, h_r$ ) at the desired value with respect to the leader aircraft. Namely, the heading velocity control  $u_v$ , heading angle control  $u_\psi$ , and altitude control  $u_h$  are to be designed so that

$$x_r = x_l - x_w \rightarrow x_d \quad (16.8)$$

$$y_r = y_l - y_w \rightarrow y_d \quad (16.9)$$

$$h_r = h_l - h_w \rightarrow h_d \quad (16.10)$$

where  $x_d, y_d$  and  $h_d$  are the desired formation distance in  $x, y$  and  $h$  coordinates. For simplicity, we only design the control algorithm for lateral ( $x$ -axis), longitudinal ( $y$ -axis) spacing, the similar control strategy could be easily applied to vertical ( $h$ -axis) spacing.

The intelligent control strategy consists of fuzzy logic control and pursuit guidance such that the Wingman could be control to achieve the formation with leader and follow the trajectory during the mission time. The automatic formation flight control can be expressed in terms of the fuzzy rules using linguistic values. The two fuzzy logic controller modules are responsible for latitude and longitude control. The inputs of the fuzzy system are heading velocity difference and heading angle difference. The output is the heading velocity and heading angle control signal. These two commands could sufficiently navigate the two aerial vehicles.

All input and output linguistic variables have a finite number of linguistic values with membership function defined after comprehensive simulation studies. The two observable flight variables are heading velocity and angle errors  $[\hat{\Delta}v, \hat{\Delta}\psi]$  and the derivatives  $[\hat{\Delta}v', \hat{\Delta}\psi']$ .

The algorithm first converts the crisp input estimates  $[\hat{\Delta}v, \hat{\Delta}\psi]$  and  $[\hat{\Delta}v', \hat{\Delta}\psi']$  to the fuzzy values  $[\hat{\Delta}v^f, \hat{\Delta}\psi^f]$  and  $[\hat{\Delta}v'^f, \hat{\Delta}\psi'^f]$ , then applies a set of inference rules to produce a fuzzy estimate of the control output  $[\hat{v}^f, \hat{\psi}^f]$  and finally computes the crisp estimate  $[\hat{v}, \hat{\psi}]$ . The fuzzy rules are defined in Tables 16.3 and 16.4.

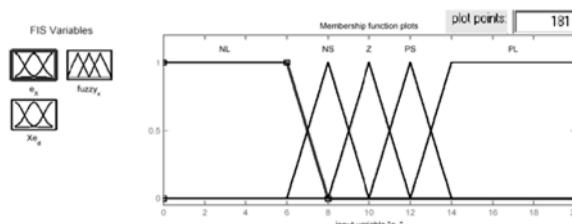
**Table 16.3.** Input of error and derivative of error

Input	Meaning
PL	Positive Large
PS	Positive Small
Z	Balance
NS	Negative Small
NL	Negative Large

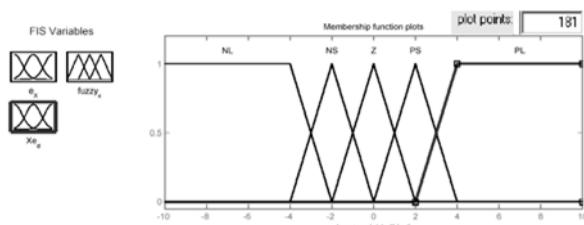
**Table 16.4.** Output of velocity and angle control

Output	Meaning
PL	Positive Large
P	Positive
O	No change
N	Negative
NL	Negative Large

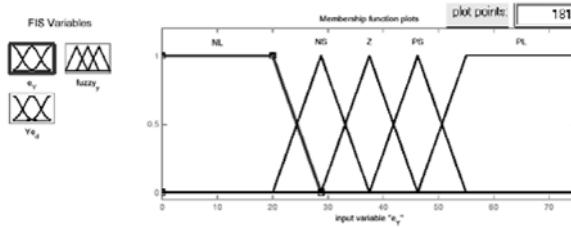
The input membership functions for velocity error and its derivative along with heading angle error and its derivative are plot in Figures 16.3 ~ 16.6.



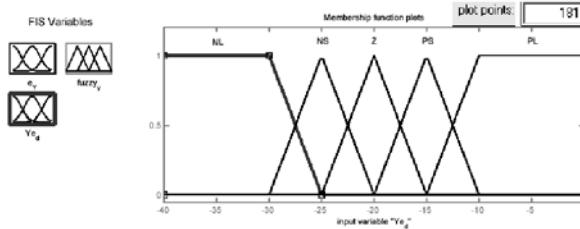
**Figure 16.3.** Membership function for velocity error



**Figure 16.4.** Membership function for derivative of velocity error



**Figure 16.5.** Membership function for heading angle error



**Figure 16.6.** Membership function for derivative of heading angle error

One of the commonly used rules of fuzzy implication is Mamdani's method associated with the max-min composition. The 'meaning' of a single rule is represented as a fuzzy relation defined as

$$\forall i : \tilde{R}_i^k = \min_{x \times y} (\mu_{A(k)}(x), \mu_{B(k)}(y)) / (x, y)$$

where  $\int_x \mu_{A(k)}(x) / x$  is the membership function of "x is  $A^{(k)}$ " ( $A^{(k)}$  is the linguistic value of the input variable X in the  $k$ th rule), similarly for  $\int_y \mu_{B(k)}(y) / y$ . The meaning of whole set of rules is defined as

$$\tilde{R}_m = \bigcup_{k=1}^n \tilde{R}_m^k,$$

which means

$$\forall x, y : \mu_{R_m}(x, y) = \max \min(\mu_{A(k)}(x), \mu_{B(k)}(y))$$

The fuzzy set  $\tilde{Y}_{out}$  is the output that corresponding to the given fuzzy input  $x^*$ , and is denoted as

$$\tilde{Y}_{out} = \mu^* \circ \tilde{R}_m = \int \mu_{B(k)}^*(y) / y$$

The defuzzifier operator is defined as

$$y^* = \frac{\int y \mu_{B(k)}^*(y)}{\int \mu_{B(k)}^*(y)}$$

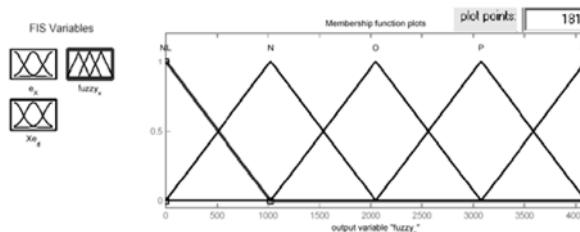
The output fuzzy rules are defined in Tables 16.5 and 16.6. The output membership functions for velocity error and its derivative along with heading angle error and its derivative are plot in Figures 16.7 and 16.8.

**Table 16.5.** Control signal of velocity

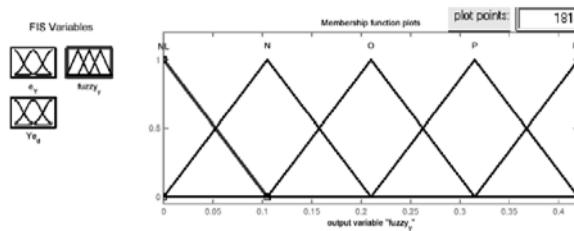
	<b>NL</b>	<b>NS</b>	<b>Z</b>	<b>PS</b>	<b>PL</b>
<b>NL</b>	O	N	P	NL	N
<b>NS</b>	P	O	PL	PL	NL
<b>Z</b>	N	NL	O	PL	PL
<b>PS</b>	N	P	NL	O	NL
<b>PL</b>	N	P	PL	NL	O

**Table 16.6.** Control signal of heading angle

	<b>NL</b>	<b>NS</b>	<b>Z</b>	<b>PS</b>	<b>PL</b>
<b>NL</b>	O	N	P	NL	N
<b>NS</b>	P	O	PL	PL	NL
<b>Z</b>	N	NL	O	PL	PL
<b>PS</b>	N	P	NL	O	NL
<b>PL</b>	N	P	PL	NL	O



**Figure 16.7.** Membership function for velocity control output



**Figure 16.8.** Membership function for heading angle control output

The Mamdani fuzzy model/control is a static MIMO system, and only returning non-zero degree of membership. A more flexible and dynamic model by using correlation-minimum approach is designed:

$$\gamma_k = \min(\lambda_1, \lambda_2),$$

$$\delta_k = \max(\lambda_1, \lambda_2),$$

where  $\gamma_k$  is the resulting degree of membership, or truth, of the fuzzy output  $\hat{v}^f$  in the fuzzy set and  $\delta_k$  is the degree of membership of the fuzzy input  $\hat{\psi}^f$  in the fuzzy set.  $\lambda_i$  is the degree of membership of the fuzzy input  $[\hat{\Delta}v^f, \hat{\Delta}\psi^f]$  in the fuzzy set. The correlation-minimum approach is selected for its ability to eliminate ambiguities in each input by only returning nonzero degrees of membership if both antecedents are nonzero. When all of the inference rules are evaluated, the degree of membership of the fuzzy output variable in each of its fuzzy sets is known as in Tables 16.5 and 16.6. The fuzzy control output is generated by using centroid computation method as [9]

$$\hat{v} = \sum_{k=1}^K \gamma_k \bar{b}_k / \sum_{k=1}^K \gamma_k ,$$

$$\hat{\psi} = \sum_{k=1}^K \delta_k \bar{b}_k / \sum_{k=1}^K \delta_k ,$$

where  $\bar{b}_k$  is the centroid location of the partially activated fuzzy output membership function. The control inputs to achieve lateral and longitudinal tracking are generated as

$$U = U_0(V, \psi, \dot{\psi}) + U(\hat{v}_f, \hat{\psi}_f, \dot{\hat{\psi}}_f)$$

This method is simple in structure and is effective in dealing with the uncertainties and unpredictable disturbances.

## 16.4 Simulations

The fuzzy navigation algorithm was integrated with a latitude and longitude formation controller in a two-aircraft simulation. A control scheme of velocity and heading angle was provided to the fuzzy system. Results show that, under the presence of input uncertainties, the fuzzy algorithm generally provides estimates accurate to within  $\pm 20$  in. The trailing aircraft was able to maintain a significant level of drag reduction while tracking the motion of the leading airplane caused by turbulence effects.

The control system based on the dynamic model given above is simulated by using MATLAB<sup>®</sup>. The aircraft characteristic values for the simulation are chosen as in Table 16.7. The flight parameters are set in Table 16.8. Table 16.9 lists the initial and designate simulation values. The simulation results including the tracking trajectory, tracking error and control gain are illustrated in Figures 16.9, 16.10 and 16.11.

**Table 16.7.** Aircraft characteristic values

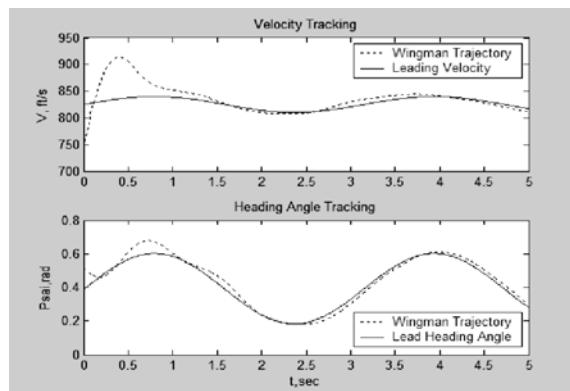
Parameter	Value	Unit
Velocity time constant	5	s
Heading time constant	0.75	s
Velocity	825	ft/s
Wing area	300	ft <sup>2</sup>
Wing span	30	ft
Mass	776.4	lb

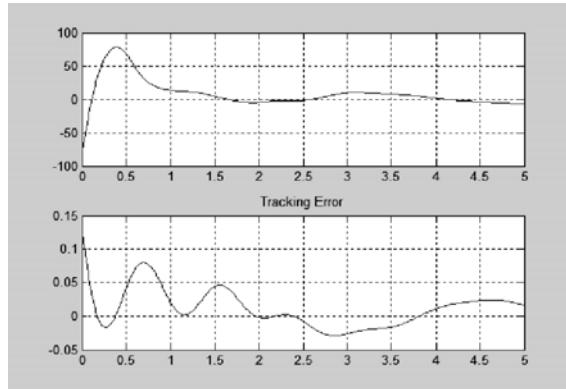
**Table 16.8.** Flight simulation parameters

Parameter	Value	Unit
Altitude	45,000	ft
Dynamic pressure	155.8	lb/ft <sup>2</sup>
Mach	0.85	Mach

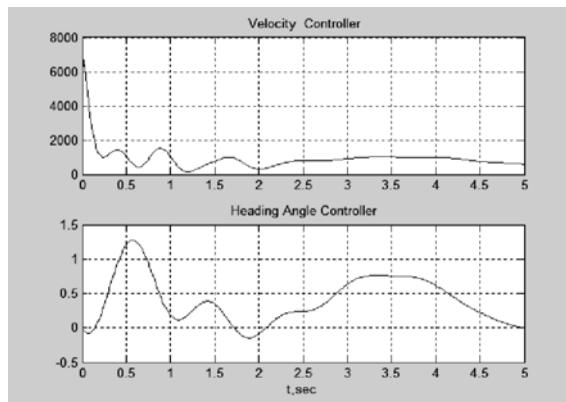
**Table 16.9.** Designate simulation values

$[x, y, z]$	$\begin{bmatrix} 100\text{ft} & 60\text{ft} & 0\text{ft} \\ 90\text{ft} & -50\text{ft} & 0\text{ft} \end{bmatrix}$
$[x_d, y_d, z_d]$	$\begin{bmatrix} 60\text{ft} & 30\pi/4\text{ft} & 0\text{ft} \\ 60\text{ft} & -30\pi/4\text{ft} & 0\text{ft} \end{bmatrix}$
$[V_w(0), \psi_w(0), h_w(0)]$	$[750\text{ft/s}, \pi/6, 30,000\text{ft}]$
$[V_L, \psi_L, h_L]$	$[825\text{ft/s}, \pi/8, 30,000\text{ft}]$

**Figure 16.9.** Tracking trajectory of a sinusoid function



**Figure 16.10.** Tracking error of the sinusoid function trajectory



**Figure 16.11.** Velocity and heading angle control input

The simulation results indicate that the proposed method maintained a desired tracking precision and also ensure the control action smooth under the system uncertainties and external disturbances.

## 16.5 Conclusion

This chapter explored a new method to achieve close formation tracking control of multi-UAVs by applying fuzzy logic theory. The formation trajectory is achieved by the control of Wingman's heading velocity and heading angle while the lead UAV is maneuvering the given mission path.

Since the values for specifying linguistic terms of the controller output for trailing UAV actually determine the shape of the input-output shape, they are selected as the main parameters for optimization. We have successfully used such a

fuzzy controller to approximate the nonlinear system, which can be applied for the supervised learning.

The method guarantees the desired tracking precision and also ensures smooth control action under system uncertainties and external disturbances. Simulation of leader-wingman formation geometry was conducted. Both theoretical studies and simulation results demonstrate that this control algorithm is simple and less sensitive to the flight aerodynamic model.

## Acknowledgement

This work was supported in part by ONR through the grant N00014-03-1-0462.

## References

- [1] Y. D. Song, Y. Li, M. Bikdash and T. Dong. "Cooperative Control of Multiple UAVs in Close Formation Flight via Smooth Robust and Adaptive Approach", *4th International Conference on Cooperative Control and Optimization*, 2003, pp 759-765.
- [2] L. E. Buzogany, M. Pachter and J. J. D'Azzo, "Automated Control of Aircraft in Formation Flight", *AIAA Guidance, Navigation, and Control Conference*, 1993, pp 1349-1370.
- [3] V. P. Reyna, M. Pachter and J. J. D'Azzo, "Formation Flight Control Automation", *AIAA Guidance, Navigation, and Control Conference*, 1994, pp 1379-1404.
- [4] A. W. Proud, M. Pachter, and J. J. D'Azzo, " Close Formation Flight Control ", *AIAA Guidance, Navigation, and Control Conference*, 1999, pp 1231-1246.
- [5] S. N. Singh, "Adaptive Feedback Linearization Nonlinear Close Formation Control of UAVs", *American Control Conference*, 2000, pp 854-858.
- [6] F. Giulitti, L. Pollini, and M. Innocenti, "Autonomous formation flight", *IEEE Control System*, Vol. 20, No. 6, 2000, pp 34-44.
- [7] R. Fierro, C. Belta, J.P. Desai and V. Kumar, "On Controlling Aircraft Formation", *40th IEEE Conf on Decision and Control*, 2001, pp 1065-1070
- [8] Kosko, Bart, *Neural Networks and Fuzzy Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1992.
- [9] Glorenne, Pierre Yves, "Learning Algorithms for Neuro-Fuzzy Networks," *Fuzzy Control Systems*, CRC Press, Inc., Boca Raton, FL, 1994.

## Applications of Fuzzy Logic in Data Mining Process

Hang Chen

### 17.1 Introduction

We all work in an information age in which data are plentiful. The demands of information management, research, and analysis have become more and more important. For example, a new customer logins into an online retailer, and places an order for several new released CDs. It will be one of hundreds of music CD orders that day. Is it likely that other customers would also like to buy these CDs? Is a new customer likely to become a repeat customer? Or is this a “once-and-done” customer? Which kinds of customers are most likely to respond to the next promotional mailing, and why?

What useful information and knowledge could we get from these orders? Can we accurately classify customers using data collected from his/her order? If we classify correctly, will this information be useful in targeted marketing? By answering these questions, we could have a better understanding of customer behavior, improve the service provided, and increase the business opportunities. On the other hand, the amount of data generated and gathered has been increasing rapidly. This explosive growth in data has generated the need for new techniques and tools that can intelligently and automatically analyze relevant data. The traditionally method were too time-consuming to cope with the massive quantities of date. As a result, data mining has become a very important research topic, which replies upon powerful computers and efficient algorithms to enhance the value of existing information resources, and can be applied to business management, government administration, scientific and engineering data management and many other areas.

Data mining is a process of nontrivial extraction of implicit, previously unknown and potentially useful information (such as knowledge rules, constraints, and regularities) from data [1]. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on-line. When implemented on high performance client/server or parallel processing computers, data mining tools can analyze massive databases to deliver answers to questions. In many articles and documents there are other terms, carrying a similar or slightly different meaning, such as knowledge discovery, data analysis, knowledge extraction, *etc.*

Chen *et al.* [1] provide a comprehensive survey of data mining. They summarized seven requirements and challenges of data mining, which included handling different types of data, efficiency and scalability of data mining algorithms, usefulness, certainty, and expressiveness of data mining results, expression of various kinds of data mining results, interactive mining knowledge at multiple abstraction levels, mining information from different sources of data, and protection of privacy and data security. Some of these requirements may carry conflicting goals. Therefore, different techniques are used to address some of these problems.

## 17.2 Foundation of Data Mining

The data mining process is composed of a few steps leading from raw data collections to some form of new knowledge. The iterative process consists of the following steps:

- **Data cleaning:** a phase in which noise data and irrelevant data are removed from the collection.
- **Data integration:** at this stage, multiple data sources, often heterogeneous, may be combined in a common source.
- **Data selection** in this step, the data relevant to the analysis is decided on and retrieved from the data collection.
- **Data transformation:** also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.
- **Data mining:** the crucial step in which algorithms and models are applied to extract patterns potentially useful.
- **Pattern evaluation:** in this step, strictly interesting patterns representing knowledge are identified based on given measures.
- **Knowledge representation:** the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.

Some steps can be combined together. Data cleaning and data integration can be performed together as a pre-processing phase to generate a data warehouse.

Data mining is an iterative process. Once the discovered knowledge is presented to the user, the evaluation measures can be enhanced, the mining can be further refined, new data can be selected or further transformed, or new data sources can be integrated, in order to get different, more appropriate results.

Data mining applications can be classified into sets of problems that have similar characteristics across different application domains. The parameterization of the application is distinct from industry to industry and application to application. The techniques and tools could be the same. For example, the model used to predict customer choice for a telephone service provider can be applied by

a cable company to predict its customer choice. The difference is how the techniques and models are parameterized. In other words, which of the domain-specific attributes should be used in the analysis and how they should be used?

Each kind of data mining is implemented by a set of techniques used to extract the useful relationships in the data. Zaiane [3] and Bruce Moxon [2] described the data mining functionalities and knowledge they discovered, which are briefly presented in the following: characterization, discrimination, association, sequence-based analysis, classification, clustering, prediction, Outlier analysis, evolution and deviation analysis, and estimation. These techniques are able to solve certain types of problems [8-10].

- **Characterization:** Data characterization is a summarization of general features of objects in a target class, and produces what are called *characteristic rules*. The data relevant to a user-specified class are normally retrieved by a database query and run through a summarization module to extract the essence of the data at different levels of abstractions. For example, one may want to characterize the on-line CD retail customers who regularly buy more than ten CDs a year.
- **Discrimination:** Data discrimination produces what are called *discriminated rules* and is basically the comparison of the general features of objects between two classes referred to as the *target class* and the *contrasting class*. For example, one may want to compare the general characteristics of the customers who bought more than ten CDs in the last year with those whose order is lower than five. The techniques used for data discrimination are very similar to the techniques used for data characterization with the exception that data discrimination results include comparative measures.
- **Association:** Association techniques cope with the problems described as market-basket analysis. Classic market-basket analysis treats the purchase of a number of items as a single transaction. For example, it could be useful for the video-rental-store manager to know what movies are often rented together or if there is a relationship between renting a certain type of movies and buying popcorn or pop. The discovered association rule often expresses the resultant item affinities in terms of confidence-rated rules, such as, “2% of the transactions considered are of customers aged between 13 and 19 who are renting a game and buying pop”, and “there is a certainty of 55% that teenage customers who rent a game also buy pop.” Confidence thresholds can typically be set to eliminate all but the most common trends. The results of the association analysis may trigger additional analysis. The goal of association rule is to find trends across large numbers of transactions that can be used to understand and exploit natural buying patterns. This information can be used to adjust inventories, modify floor or shelf layouts, or introduce targeted promotional activities to increase overall sales or move specific products. While these approaches had their origins in the retail industry, they can be applied equally well to services that develop targeted marketing campaigns or determine common (or uncommon) practices. In terms of

financial analysis, association approaches can be used to analyze customers' account portfolios and identify sets of financial services that people often purchase together. They may be used, for example, to create a service "bundle" as part of a promotional sales campaign.

- **Sequence-based analysis:** Traditional market-basket analysis deals with a collection of items as part of a point-in-time transaction. A variant of this problem occurs when there is additional information to tie together a sequence of purchases (for example, an account number, a credit card, or a frequent buyer/flyer number) in a time series. In this situation, not only may the coexistence of items within a transaction be important, but also the order in which those items appear across ordered transactions and the amount of time between transactions. Sequence-based analysis can be used, for example, to identify a typical set of precursor purchases that might predict the subsequent purchase of a specific item.
- **Classification:** Classification analysis is the organization of data in given classes. Classification employs a training set where all objects are already classified with known class labels to develop a model that can classify the population of records at large. Fraud detection and credit-risk applications are particularly well suited to this type of analysis. This approach frequently employs decision tree or neural network-based classification algorithms. The use of classification algorithms begins with the training set of pre-classified objects transactions. For a fraud detection application, this would include complete records of both fraudulent and valid activities, determined on a record-by-record basis. The classifier training algorithm uses these pre-classified examples to determine the set of parameters required for proper discrimination. The classification algorithm learns from the training set and builds a model called a classifier. For example, after starting a credit policy, a video rental store manager could analyze the customers' behaviors via their credit, and label accordingly the customers who received credits with three possible labels "safe", "risky" and "very risky". The classification analysis would generate a model that could be used either to accept or to reject credit requests in the future.
- **Clustering:** Clustering approaches address segmentation problems. These approaches assign records with a large number of attributes into a relatively small set of groups or "segments." Similar to classification, clustering is the organization of data in classes. However, unlike classification, in clustering, class labels are unknown and it is up to the clustering algorithm to discover acceptable classes. There are many clustering approaches all based on the principle of maximizing the similarity between objects in a same class (*intra-class similarity*) and minimizing the similarity between objects of different classes (*inter-class similarity*). Clustering is often one of the first steps in data mining analysis. It identifies groups of related records that can be used as a starting point for exploring further relationships. This technique supports the development of population segmentation models, such as demographic-based customer segmentation. Additional analyses using

standard analytical and other data mining techniques can determine the characteristics of these segments with respect to some desired outcome. For example, the buying habits of multiple population segments might be compared to determine which segments to target for a new sales campaign.

- **Estimation:** A variation on the classification problem involves the generation of scores along various dimensions in the data. Rather than employing a binary classifier to determine whether a loan applicant is a good or bad risk, this approach generates a credit-worthiness "score" based on a pre-scored training set.
- **Prediction:** There are two major types of predictions. One can either try to predict some unavailable data values or pending trends, or predict a class label for some data. The other one is tied to classification. Once a classification model is built based on a training set, the class label of an object can be foreseen based on the attribute values of the object and the attribute values of the classes. Prediction however more often refers to the forecast of missing numerical values, or increase/ decrease trends in time related data. The major idea is to use a large number of past values to consider probable future values. Prediction has attracted considerable attention given the potential implications of successful forecasting in a business context.
- **Outlier analysis:** Outliers are data elements that cannot be grouped in a given class or cluster. Also known as *exceptions* or *surprises*, they are often very important to identify. While outliers can be considered noise and discarded in some applications, they can reveal important knowledge in other domains, and thus can be very significant and their analysis valuable.
- **Evolution and deviation analysis:** Evolution and deviation analysis pertain to the study of time related data that changes in time. Evolution analysis models evolutionary trends in data, which consent to characterizing, comparing, classifying or clustering of time related data. Deviation analysis, on the other hand, considers differences between measured values and expected values, and attempts to find the cause of the deviations from the anticipated values.

There are other data mining techniques including case-based reasoning, genetic algorithms, and fuzzy logic. Each of these has its own strengths and weakness in terms of problem types addressed, performance and complexity. In many applications, which kind of patterns can be discovered or need to be discovered from the data are not determined at hand. It is therefore very important to have a versatile and inclusive data mining system, which can discover different kinds of knowledge at different levels of abstraction. This also makes interactivity an important attribute of a data mining system.

The technologies studied in this area have mainly been focused on highly structured and precise data. In addition, some of these techniques are highly mathematical and are quantitative in nature. The goal to obtain understandable results is often ignored. To exploit fully all the attributes of an object present in the

data set, one must use the qualitative attributes. The analysis of heterogeneous information sources with the prominent aim of producing comprehensible results is a new challenge in data mining research. Fuzzy logic is an extraordinarily valuable tool for representing and manipulating all kinds of data in qualitative/linguistic terms and for achieving understandable solutions. In this chapter, we will introduce the basic concepts of fuzzy logic, explain its advantages over conventional techniques and describe the models which use fuzzy logic in data mining, and explore the possible research topics of data mining using fuzzy logic method.

### 17.3 Fuzzy Logic for Data Mining Basic

It is undisputed that language is a most effective human tool to structure experience and to model environment. It is therefore necessary to model linguistic terms. What Zadeh proposed as computing with words [7] indicated a new direction in data mining technologies. Linguistic terms are vague in nature, *i.e.*, they have “fuzzy” boundaries. For each linguistic term there are some phenomena to which it can clearly be applied and others which cannot be described using this term. However, there lies a “penumbra” of phenomena for which it is not definite whether the term is applicable or not.

The reason for this inherent vagueness is that for practical purpose full precision is not necessary and may even be a waste of resources. Fuzzy set theory provides excellent tools to model the “fuzzy” boundaries of linguistic terms by introducing gradual membership. In classical set theory, an object is either a member of a given set or not. Fuzzy set theory makes it possible that an object or a case belongs to a set only to a certain degree, thus modeling the penumbra of the linguistic term describing the property that defines the set.

Member degrees of fuzzy sets include similarity, preference, and uncertainty. They can state how similar an object or case is to a prototypical one, they can indicate preferences between suboptimal solutions to a problem, or they can model uncertainty about the real life situation if the scenario is described in an imprecise manner. Due to their closeness to human reasoning, a solution obtained using fuzzy approaches is easy to understand and to apply. Fuzzy systems are therefore good candidate to choose, if linguistic, vague, or imprecise information has to be modeled and analyzed.

Recently, there has been a suggestion for general data modeling, which indicated that data mining is a circular process. The evaluation of the data mining result can trigger a re-execution of the data preparation and model generation steps. As a process, fuzzy set methods can be applied in several phases: [15, 16]

- **Problem understanding phases:** These phases serve mainly to identify the goals of the data mining task, and to gather and select required data. In these phases, fuzzy set methods can be used to formulate, for example, the background domain knowledge in vague manner, which can be used for the subsequent modeling phase. Moreover, fuzzy database queries are useful for finding the data needed and to check whether it may be useful to take additional related data into consideration.

- **Data preparation step:** The data collected is cleaned, transformed to generate the input for modeling. Fuzzy method can be used to detect outliers. For example, fuzzy clustering the data and then finding those data points that far away from the cluster prototypes. Moreover, selecting and extracting useful attributes of target objects from most of database are difficult. Not all of the attributes needed for successful extraction can be contained in the database. In these cases, domain knowledge and user analysis becomes a necessity. In these cases, techniques such as neural networks tend to do badly since the domain knowledge cannot be incorporated into the neural networks (Lu *et al.* 1996) [14]. Fuzzy logic based models utilize the domain knowledge in coming up with rules of data selection and extraction.
- **Modeling phase:** Models are constructed in this phase. Fuzzy data analysis approaches can be applied to predict future developments or to build classifiers. One kind of application is to analyze fuzzy data, which are derived from imprecise measurement instruments or from the descriptions of human domain experts. The other kind of applications consists of methods that use fuzzy techniques to structure and analyze crisp data. As described in the introduction, data mining is the process of extracting nontrivial relationships in the database. However, associations that are qualitative are very difficult to utilize effectively by applying conventional rule induction algorithms. Since fuzzy logic modeling is a probability based modeling, it has many advantages over the conventional rule induction algorithms. These advantages are summarized in Maeda *et al.* (1995) [12]. The first advantage is that it allows processing of very large data sets which require efficient algorithms. Fuzzy logic-based rule induction can handle noise and uncertainty in data values well.
- **Evaluation phase:** In this phase, results are tested and assessed. Fuzzy modeling methods are interpretable systems. Therefore, they can easily be checked for plausibility against the intuition and expectations of human experts. The results can also provide unknown knowledge for new consideration.

To illustrate how to apply fuzzy data analysis method in data mining process, in the following section we introduce some example in a little more details.

## 17.4 Algorithms of Fuzzy Logic in Data Mining Process

The two projects described by Nakajima [11] explained actual applications of data mining utilizing fuzzy logic both from development point of view and also from a conceptual point of view.

The first problem is to identify customers from a customer database who can be targeted for a sale under the assumption that these customers responded positively to advertisements mailed to them. The additional constraint is that the mailing list budget is limited and the number of advertisements to be mailed is to be controlled to increase profit. The first step involves analyzing the database for attributes like

"frequency of visits to the store", "sum of purchases", *etc.* Analysis and plots of the data then determine the cluster of good customers. Next, one has to find the attribute relationships to define a query condition, which is represented by a pair of attributes and a fuzzy linguistic value. One then verifies and refines the query condition by using another customer database. Thus the customer database is ranked and sorted by degree values based on a given fuzzy query condition. The customers retrieved by the query determine the list of the potential of good customers.

In the other project, a product that was used to sense vibrations and predict the causes of these vibrations (*i.e.* earthquakes, *etc.*) was improved by utilizing fuzzy rules. The original sensor was based on a simple threshold rule. The error rate for this sensor was around 12%. The fuzzy rules were created by analyzing the actual data in specified cases of earthquakes, automobiles *etc.* A feature extraction was done on the data set to identify each kind of causes. Relationships between the feature parameters and the kind of vibration were discovered to develop the fuzzy rules. These rules were then tested and refined. The accuracy of the sensor improved dramatically with the error rate falling to within 1%. The following describes several simple fuzzy logic algorithms.

#### *Rule Extraction: Wang-Mendel Algorithm*

The first application is to generate rules from data. The algorithm proposed by Wang and Mendel [5, 6] generates the set of rules in a fuzzy system based on a table of data with  $n$  dimension, where  $n-1$  dimensions are used as input and the last dimension describes the output required from the fuzzy system. The basic idea is that each tuple in the dataset, containing many input variables and one output variable, represents a rule. The steps are:

- Divide the input and output spaces into fuzzy regions: it's a fuzzification step. In this step, it incorporates dividing each domain in  $2N+1$  regions, and assigns a fuzzy membership function to each region. There is overlapping between each two adjacent intervals. The value of  $N$  and the regions or the membership functions do not need to be the same for all domains. The authors proposed triangular membership functions, and suggested that the other shapes could also work. The fuzzy membership functions are named as SMALL N (SN), ...SMALL1 (S1), CENTER (C), LARGE1 (L1), ..., LARGE N (LN).
- Generate fuzzy rules from given tuples: Every tuple is transformed into an IF-THEN rule, using the best IF-THEN rule that describes it, which is the rule made by the regions of maximum degree for each value of the tuple. For example, suppose that in  $(x_1, x_2, y)$  the value of  $x_1$  can be described as  $0.4/X2+0.5/X1$ , while the value of  $x_2$  is  $0.3/B2+0.6/B1$  and  $y$  is  $1/C$ . The rule should then be: IF  $x_1$  is  $X1$  AND  $x_2$  is  $B1$  THEN  $y$  is  $C$ .
- Assign a degree to each rule: The dataset considered is usually large. Each pair of data generates one rule. We therefore need a method of choosing from among two or more conflicting rules. Two rules will be

in disagreement if they have the same antecedent but different consequent. Values are assigned for each rule, which is called the degree of the rule. The rule with the highest degree from among all conflicting rules is selected. The degree of a rule is the product of all membership values described in the rule. The tuples could have some prior information about its quality as a representative result of the system. The information can be described as a number. The degree of the function can be multiplied by this value.

- Create the combined set of rules: In this step, all conflicting rules but the ones with highest degree are eliminated to guarantee a non-conflicting set of rules.
- Determine the mapping based on the set of rules: The final step defines the processing and defuzzification method for the resulting fuzzy system. The product of the membership degrees of the antecedent is used as an implication method to determinate the membership degree of the consequent. Finally, it uses the center of sums defuzzification method.

### *Genetic Algorithm*

Genetic algorithms have gained increasing attentions. Genetic algorithms are designed to find a global maximum of a function of many variables by performing a particular kind of genetic-inclined search in the space of these variables. This form of search is especially interesting when solving complex optimization tasks, including the structure and parameters optimization problem. The main principle of a genetic algorithm is described as follows.

It begins with an initial set (also called population) of randomly generated potential solutions to an optimization problem. The value of a fitness function is evaluated for each solution, and the "best" solutions are selected for survival according to the selection probabilities. For each solution, the selection probability is defined as a function of its fitness value. Then, the genetic algorithm manipulates these selected solutions in its search for better solutions. Each solution is encoded into a binary string, so that new encoded solutions can be generated through the exchange of information among surviving solutions (crossovers) as well as sporadic alternations in the bit string encodings of the solutions (mutations). The power of genetic algorithms is in their capacity of simultaneously exploring several regions in the space of potential solutions. They can be run without a learning base. During the running of a genetic algorithm, the search concentrates progressively on the regions where the optimum is located in.

Wang and Mendel's algorithm uses each tuples as a candidate for a single rule. In [13], Goldberg proposed a genetic algorithm, which searches the space of all possible set of rules for the specific set of rules that better represent the data as whole. In [10], Xexéo describe the algorithm as five steps:

- Select linguistic variables and linguistic terms
- Find a suitable representation for the genotype
- Select an evaluation function

- Generate randomly a large number of complete sets of rules
- Run the genetic algorithm

The author also uses a simple example to describe how the algorithm works.

**Table 17.1.** Example of dataset for the function  $Z=X+Y$

X	Y	Z
0	0	0
0	1	1
0	2	2
1	0	1
1	1	2
1	2	3
2	0	2
2	1	3
2	2	4

Using linguistic terms (Small, OK, Big) for each linguistic variable ( $x, y, z$ ), the set of rules can be represented as a  $3 \times 3$  matrix where each cell can assume one of three values. Use two bits to represent each cell value. For instance, 00 for Small, 01 for OK, 11 for Big, and read the matrix left to right, top to bottom. The matrix is represented as (000001000111011111).

**Table 17.2.** Example of possible rules

	Small	OK	Big
Small	Small	Small	OK
OK	Small	OK	Big
Big	OK	Big	Big

The third step is to select an evaluation function. A good one will provide information about how well the set of rules worked as a predictor. The mean square error of every prediction is a good choice.

The fourth step is to generate solutions randomly. The last step is to apply the genetic algorithm, using the evaluation function defined in the previous step to define the survival of elements. The best solution could be selected among the population when the criterion of convergence is satisfied.

### *Dependency Analysis*

Dependency analysis is a heuristic search technique for finding significant relationships among attributes in large datasets. Reasoning in high dimension domains becomes infeasible due to the exponential scaling of computational load with respect to the number of attributes considered. Moreover, uncertainty and imprecision involved make the task even more difficult to cope with. In [4], Kruse,

Borgelt, and Nauck describe a probabilistic reasoning approach. Decomposition techniques are applied to simplify the reasoning process. Decomposition is based on dependence and independence relations between the attributes or variables that are used to describe the domain under consideration. The structure of these dependence and independence relation are represented as graph, in which each node stands for an attributes and each edge for a direct dependence between two attributes. The precise set of dependence and (conditional) independence statements can be presented in the graph using simple graph theoretic criteria. The conditional independence graph is only the qualitative or structural components of a graphical model. To do reasoning, it has to be enhanced by a quantitative component that provides confidence information about the different points of the underlying domain. This information can often be represented as a distribution function on the underlying domain, for example, a probability distribution. The conditional independence graph describes a factorization of the distribution function on the domain as a whole into conditional or marginal distribution functions on lower-dimensional subspaces. Such a graphic model makes reasoning much more efficient.

The standard approach to construct a graphical model has been to let a human domain expert specify the dependency structure of the considered domain, which provided the conditional independence graph. The human domain expert then estimated the necessary conditional or marginal distribution functions, which formed the quantitative components of the graphical model later. If there is no or only vague knowledge available about the dependence and independence relations in the domain, this algorithm become very inefficient. Moreover, it is a time-consuming approach considering the exponential scaling of computational load with respect to the number of attributes. Therefore, using sample cases of databases to learn graphical has been discussed recently [17-19].

## 17.5 Conclusion

As described in the introduction, data mining is the process of extracting nontrivial relationships from data. No single technique can be defined as the optimal technique for data mining. Fuzzy logic modeling is a probability based method. It has many advantages over the conventional rule induction algorithm, which has been discussed in this chapter. The selection of technique depends on the problem and the data set. How to apply fuzzy logic to data mining problems is still a relatively new field. There are great potentials for exploiting these techniques on different data sets to extract information.

## References

- [1] M. S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database perspective. *IEEE Trans. Knowledge and Data Engineering*, 8:866-883, 1996.
- [2] Bruce Moxon. Defining Data Mining. *DBMS Data Warehouse Supplement*, August 1996.

- [3] Osmar R. Zaïane, Simeon J. Simoff, Chabane Djeraba, Mining Multimedia and Complex Data, published by Springer Verlag, Lecture Notes in Artificial Intelligence Volume 2797, 2003, ISBN:3-540-20305-2.
- [4] R. Kruse, C. Borgelt, and D. Nauck. Fuzzy Data Analysis: Challenges and Perspectives. Proc. 8th IEEE International Conference on Fuzzy Systems, 1999.
- [5] L.-X. Wang and J.M. Mendel, Generating Fuzzy Rules from Numerical Data, with Applications Anonymous USC-SIPI REPORT #169, 1991. Department of Electrical Engineering-Systems. University of South California.
- [6] L.-X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. IEEE Trans. Syst., Man, Cybern. 22:1414–1227. IEEE Press, Piscataway, NJ, USA 1992
- [7] L.A. Zadeh. Fuzzy Logic = Computing With Words. IEEE Transactions on Fuzzy Systems 4:103–111. IEEE Press, Piscataway, NJ, USA 1996
- [8] Kurt Thearling, An Introduction to Data Mining: Discovering hidden value in your data warehouse. [www.thearling.com](http://www.thearling.com)
- [9] E. Cox. *Fuzzy Logic for Business and Industry*, Rockland, Massachusetts: Charles River Media, 1995.
- [10] Geraldo Xexéo, Fuzzy Logic, teaching notes.
- [11] Nakajima H.:Fuzzy Logic and Data Mining, Soft. Computing in Intelligent Systems and Information Processing, Proceedings of the 1996 Asian Fuzzy Systems, 1996.
- [12] Maeda, A., Ashida, H., Taniguchi, Y. and Takahashi, Y. Data Mining System using Fuzzy Rule Induction, Proceedings of 1995 IEEE International Conference on Fuzzy Systems, 1995.
- [13] D.E. Goldberg. Genetic algorithms in search, optimization, and *machine learning*, Reading, Mass: Addison-Wesley Pub. Co., 1989.
- [14] Lu, H., R. Setiono, and H. Liu, Effective Data Mining Using Neural Networks, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6. 1996.
- [15] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.
- [16] W. J. Frawley, G. Piatetsky-Shapiro and C. J. Matheus, Knowledge Discovery in Databases: An Overview. In G. Piatetsky-Shapiro *et al.* (eds.), *Knowledge Discovery in Databases*. AAAI/MIT Press, 1991.
- [17] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
- [18] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. Communications of ACM, 39:58-64, 1996.
- [19] G. Piatetsky-Shapiro, U. M. Fayyad, and P. Smyth. From data mining to knowledge discovery: An overview. In U.M. Fayyad, *et al.* (eds.), *Advances in Knowledge Discovery and Data Mining*, 1-35. AAAI/MIT Press, 1996.

## **Fuzzy Logic Control for Power Networks: A Multilayer Fuzzy Controller**

Ahmed Rubaai and Abdul R. Ofoli

### **18.1 Introduction**

Transient stability is one of the most important factors studied in power system planning and operation. It is concerned with a system's ability to remain in synchronization following a sudden and major disturbance such as a generator trip or line switching due to faults, or abrupt changes in load generation powers. Several control methods have been devised to handle this problem with some conventional methods being load shedding [1], dynamic braking [2], thyristor-controlled series compensator [3], and steam valving and excitation *via* Hamiltonian function method [4, 5]. The conventional approach often requires a precise mathematical model of the controlled systems. In power system practice, there exist parameter uncertainty problems in the plant modeling and, also, since it is large, complex, geographically widely distributed, and can be affected by unexpected events, the conventional controllers often perform satisfactorily over a rather limited range of operation. These reasons also make global control very difficult.

To create flexibility and a strategy that takes into account uncertainties, a solely fuzzy logic control scheme is desirable. Unlike most of the conventional and modern controllers, fuzzy logic controllers are capable of tolerating uncertainty and imprecision to a greater extent. Hence, they produce good results under changing operating conditions and uncertainties/imprecision in system parameters. Fuzzy control is becoming a powerful control tool in the power industry with application to series-connected FACTS devices, wind–diesel power systems, photovoltaic power systems, and thyristor control to the power system stability problem [6-9]. In [15] it was argued that fuzzy control and other allied techniques such as self-organizing fuzzy control provide an alternate paradigm to analytic control theory.

During the past several years, hierarchical approaches have emerged as one of the most attractive area for research in the application of fuzzy set theory [10-14]. Several hierarchical approaches have been directed to design a series of hierarchical fuzzy processors with a small number of input variables distributed in each processor [13, 14]. Other works were based on the development of a hierarchical system of linguistic rules learning methodology [10-12]. This methodology has been thought of as a refinement of simple linguistic models

which, preserving their descriptive power, introduces small changes to increase their accuracy. In this fashion, the knowledge base structure of linguistic fuzzy-rule-based systems has been extended by introducing the concept of layers.

In this chapter, a robust hierarchical fuzzy-logic-based dynamic braking resistor controller has been implemented for a multimachine power system subjected to large disturbances. As an extension to fuzzy control, the hierarchical fuzzy controller has two layers. The first layer, termed the supervisory layer, specifies the region of operation of the subcontrollers (SCs) within the second layer, called the execution layer. The outputs of the SCs are then fuzzily combined to achieve the overall objective of the system. Following a disturbance, the generator speed deviation and its derivative are computed from real time measurements as inputs to the fuzzy logic controller and the output of the fuzzy SCs represents the conductance value of the dynamic braking resistor. The proposed controller was found to be very robust to any changes in the network configuration.

## 18.2 Description of Fuzzy Logic Systems

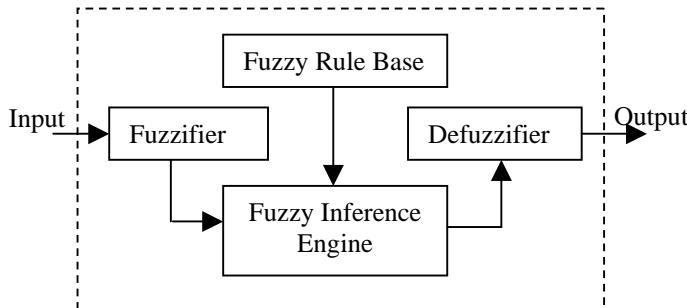
Fuzzy logic systems belong to the category of computational intelligence techniques. One advantage of fuzzy logic over other forms of knowledge-based controllers lies in the interpolative nature of fuzzy control rules. The overlapping fuzzy antecedents to the control rules provide transitions between the control actions of different rules. Because of this interpolative quality, fuzzy controllers usually require far fewer rules than other knowledge-based controllers. In a fuzzy logic controller, the dynamic behavior of a fuzzy system is characterized by a set of linguistic description rules based on expert knowledge usually of the form:

*IF (a set of conditions are satisfied) THEN (a set of consequences can be inferred)*

The premise defines a region of the input space wherein the rule is valid while the consequent defines the desired control action to be taken for the input space delimited by the premise. Figure 18.1 shows a basic configuration of a fuzzy logic system. The inputs, which are the state variables, characterize the key features of the system, and the output control variable influences the states of the system. The basic consideration is aimed at stabilizing the rotor angles of the machines and also damping the oscillation of the generator speed to zero. Following the definitions of the input and output variables and the rules, the variables are then transformed into respective fuzzy variables according to the chosen membership functions. The fuzzy variables are then expressed by linguistic terms such as “Negative Big (NB),” “Negative Medium (NM),” “Negative Small (NS),” “Zero (ZE),” “Positive Small (PS),” “Positive Medium (PM),” and “Positive Big (PB)” for the input and output parameters.

Finer adjustments of the controller output can be achieved with more rules and/or fuzzy sets. Also, with increasing number of system variables, the number of rules becomes an exponential function of the number of variables. As the number of rules increases, the management of the rules becomes more involved and it may

become unproductive to realize a fuzzy rule-based controller. To overcome this problem, a hierarchical fuzzy control structure is used where the most influential parameters are chosen as the system variables in the first layer called the supervisory layer. The constant consequent in the layer is replaced with a fuzzy controller that can cover a larger region of operation than a constant and hence reduce the number of rules. In the second layer of the MLFC, there are two execution layers running parallel to each other, each of them having two system variables as inputs to reduce the multidimensionality of the overall system.



**Figure 18.1.** Basic configuration of a fuzzy logic system

### 18.3 Functional Design of the Proposed Controller

A block diagram of the MLFC is shown in Figure 18.2. In this structure the *supervisory layer* determines the region of operation of the controller based on the rate of change of the speed (acceleration) of the machines. In effect, it determines membership values for the various sub-controllers (SCs) within the execution layer. Sub-controllers with zero or negligible membership values are not evaluated within the execution layer to increase the execution time. The implementation of the supervisory layer is done by a set of fuzzy decision rules. A typical fuzzy rule in the supervisory layer has the form

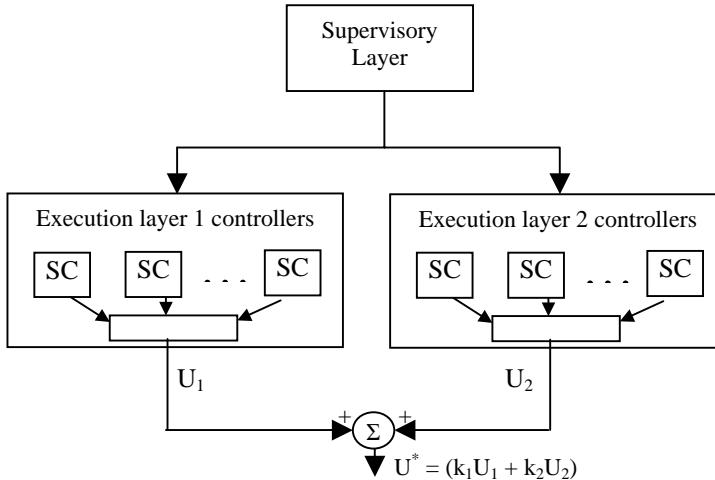
$$\text{IF } a_i \text{ THEN } SC_i$$

where the term  $a_i$  is a fuzzy expression specifying the conditions that the  $i$ th SC is relevant.

In the execution layer, the control problem is divided into smaller sub-problems which are easier to solve. Sub-controllers (SCs) are then designed to handle each of the sub-problems within the execution layer. All the sub-controllers are fuzzy in this work with a constant output but a non-fuzzy sub-controller can also be used if it is more suitable for a particular task or region of operation. The output of the execution layer is then evaluated using a fuzzy centroid method as

$$U = \frac{\sum_i^n \mu_{a_i} SC_i}{\sum_i^n \mu_{a_i}} \quad (18.1)$$

where  $\mu_{a_i}$  is the membership value of  $a_i$  determined in the supervisory layer and  $SC_i$  is the output of the  $i$ th sub-controller (SC).



**Figure 18.2.** Block diagram of the proposed MLF controller

In the control scheme, since the speed of the machine is one of the most influential parameters of the system, it was divided into five regions based on its rate of change (*i.e.* acceleration). There are five SCs to handle each region within the execution layer. The fuzzy rules used in the supervisory are given below for the fuzzy variable Acc (*i.e.* acceleration):

- R1: IF Acc is High\_Pos (HP) THEN SC\_1
- R2: IF Acc is Low\_Pos (LP) THEN SC\_2
- R3: IF Acc is Const. (ZE) THEN SC\_3
- R4: IF Acc is Low\_Neg (LN) THEN SC\_4
- R5: IF Acc is High\_Neg (HN) THEN SC\_5

where SC\_1, ..., and SC\_5 are all fuzzy controllers for the indicated regions of operation in all supervisory rules. The membership functions of the fuzzy variable Acc are shown in Figure 18.3.

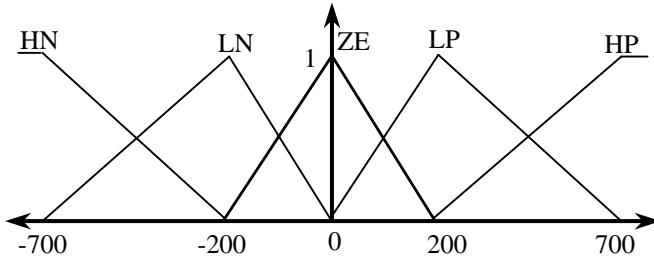


Figure 18.3. Membership set for the variable Acc

In Figure 18.3, there is a large overlap between the ‘Low’ and ‘High’ fuzzy sets to ensure smooth transition between the sub-controllers in these regions.

As shown in Figure 18.2, there are two different execution layers running parallel to each other. This structure was implemented to reduce the total number of input parameters from four to two when using the two execution layers. Execution layer 1 has generator speed deviation ( $\Delta\omega_{er}$ ) and the rate of change of the speed ( $\Delta\dot{\omega}$ ) as input parameters from the system. Execution layer 2 has the rotor angle deviation ( $\Delta\delta_{er}$ ) from the desired and the rate of change of the rotor angle ( $\Delta\dot{\delta}$ ) as input parameters from the system. The speed error  $\Delta\omega_{er}$  can be computed from real-time measurement and the set-point,  $\omega_o$ , as

$$\Delta\omega_{er}(kT) = \rho_3(\omega(kT) - \omega_o) \quad (18.2)$$

where  $\omega$  and  $\omega_o$  are measured speed and set point respectively. Then  $\Delta\dot{\omega}$  can be computed as

$$\Delta\dot{\omega}[(k+1)T] = \rho_4 \left( \frac{\Delta\omega[(k+1)T] - \Delta\omega(kT)}{T} \right) \quad (18.3)$$

where  $T$  is the sampling interval. The rotor angle deviation  $\Delta\delta_{er}$  and its rate of change  $\Delta\dot{\delta}$  are also computed as follows:

$$\Delta\delta_{er}(kT) = \rho_1(\delta(kT) - \delta_d) \quad (18.4)$$

$$\Delta\dot{\delta}[(k+1)T] = \rho_2 \left( \frac{\Delta\delta[(k+1)T] - \Delta\delta(kT)}{T} \right) \quad (18.5)$$

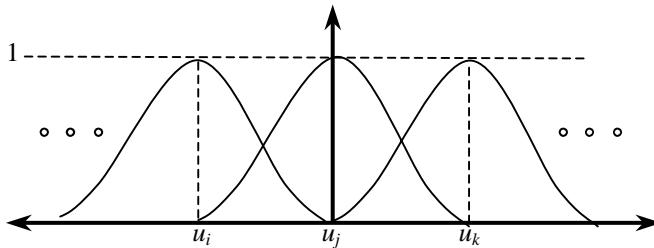
where  $\delta(t)$  is computed from the system dynamics and  $\delta_d$  is a desired value chosen for the rotor angle of the machines. Also  $\rho_1$ ,  $\rho_2$ ,  $\rho_3$  and  $\rho_4$  are scaling factors. Each execution layer produces a control signal which are weighted and then added together to yield an overall control signal,  $U^*$ , representing the conductance value of the braking resistance. That is

$$U^* = k_1 U_1 + k_2 U_2 \quad (18.6)$$

where  $k_1$  and  $k_2$  are constants values to be determined. In choosing the constants, the average value of all the membership values for the SCs in execution layer 1 was chosen to be  $k_1$  while  $k_2$  was chosen to be the average value of all the membership values for the SCs in execution layer 2. The input membership set of the input variables for the SCs are of the normal distribution form as shown in Figure 18.4 with the expression

$$\mu_{R_i}(x) = \exp\left\{-\frac{(x - u_i)^2}{2}\right\} \quad (18.7)$$

where, if the normalized universe is changed, the parameters  $u_i$  should be changed accordingly. The sets of particular interest are the membership functions for  $u$ . The fuzzy sets defined for the SCs are {NM, NS, ZE, PS, PM}.



**Figure 18.4.** Membership set for the input variables

The center of gravity for the fuzzy variables  $\Delta\omega$  and  $\Delta\delta$  are  $\{-0.80, -0.20, 0.00, 0.20, 0.80\}$  while that for the fuzzy variables  $\Delta\dot{\omega}$  and  $\Delta\dot{\delta}$  are  $\{-1.50, -0.50, 0.00, 0.50, 1.50\}$ . The membership set for the fuzzy output variable is comprised of the singletons set  $\{-151.0, -10.2, 0.0, 3.2, 50.7\}$  corresponding to the fuzzy sets {NM, NS, ZE, PS, PM} respectively.

The rules for SC\_1 for the region of the high positive acceleration are shown in Table 18.1. In this region, the speed trajectory will keep increasing and if there happens to be an overshoot, the trajectory will try to catch up with the machine speed without the need to slow down the machine quickly. The sub-controller for the region of low positive acceleration, SC\_2 is similar to that of SC\_1 with a slight modification of the rules as shown in Table 18.2. In the region where the acceleration is constant, the rules for SC\_3 are similar to a normal regulator that is shown in Table 18.3. Finally, in the region of high negative acceleration the trajectory keeps decreasing and hence the rules for SC\_5 tilt towards outputs of negative polarity. This is similar to the rules of SC\_4 but with slight difference. These are shown in Tables 18.4 and 18.5 respectively.

**Table 18.1.** Fuzzy rules for SC\_1

		$\Delta\dot{\omega}$				
		NM	NS	ZE	PS	PM
$\Delta\omega_{er}$	PM	NS	ZE	PS	PS	PM
	PS	ZE	ZE	PS	PS	PM
	ZE	ZE	PS	PS	PM	PM
	NS	PS	PS	PS	PM	PM
	NM	PS	PS	PM	PM	PM

**Table 18.2.** Fuzzy rules for SC\_2

		$\Delta\dot{\omega}$				
		NM	NS	ZE	PS	PM
$\Delta\omega_{er}$	PM	NM	NS	ZE	PS	PM
	PS	NS	ZE	PS	PS	PM
	ZE	ZE	PS	PS	PM	PM
	NS	PS	PS	PS	PM	PM
	NM	PS	PS	PM	PM	PM

**Table 18.3.** Fuzzy rules for SC\_3

		$\Delta\dot{\omega}$				
		NM	NS	ZE	PS	PM
$\Delta\omega_{er}$	PM	NM	NM	NS	ZE	PS
	PS	NM	NS	NS	PS	PS
	ZE	NM	NS	ZE	PS	PM
	NS	NS	NS	PS	PM	PM
	NM	NS	ZE	PS	PM	PM

Similar rules were used for the input variables  $\Delta\delta_{er}$  and  $\Delta\dot{\delta}$ . The values of the scaling factors used for the input variables  $\Delta\delta_{er}$ ,  $\Delta\dot{\delta}$ ,  $\Delta\omega_{er}$  and  $\Delta\dot{\omega}$ , given by  $\rho_1$ ,  $\rho_2$ ,  $\rho_3$  and  $\rho_4$  are 0.90, 0.30, 0.051, and 0.001, respectively.

**Table 18.4.** Fuzzy rules for SC\_4

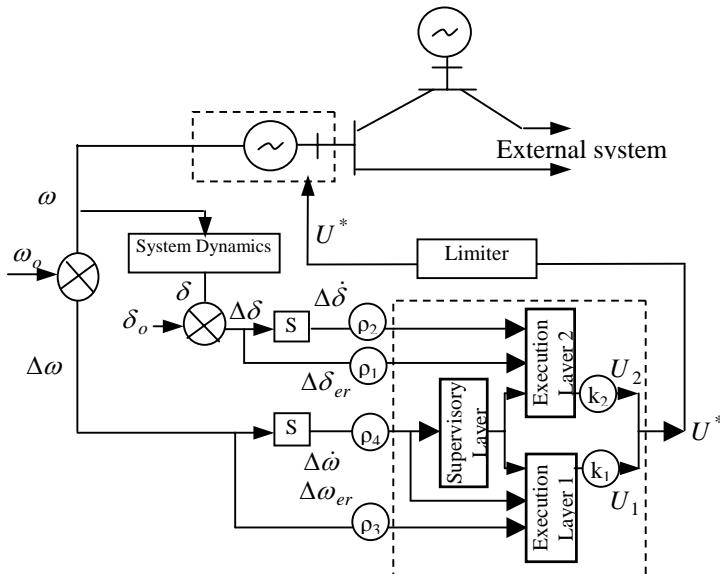
		$\Delta\dot{\omega}$				
		<b>NM</b>	<b>NS</b>	<b>ZE</b>	<b>PS</b>	<b>PM</b>
$\Delta\omega_{er}$	<b>PM</b>	NM	NM	NM	NS	NS
	<b>PS</b>	NM	NM	NS	NS	ZE
	<b>ZE</b>	NM	NM	NS	NS	ZE
	<b>NS</b>	NM	NS	NS	ZE	PS
	<b>NM</b>	NM	NS	ZE	PS	PM

**Table 18.5.** Fuzzy rules for SC\_5

		$\Delta\dot{\omega}$				
		<b>NM</b>	<b>NS</b>	<b>ZE</b>	<b>PS</b>	<b>PM</b>
$\Delta\omega_{er}$	<b>PM</b>	NM	NM	NM	NS	NS
	<b>PS</b>	NM	NM	NS	NS	NS
	<b>ZE</b>	NM	NM	NS	NS	ZE
	<b>NS</b>	NM	NS	NS	ZE	ZE
	<b>NM</b>	NM	NS	NS	ZE	PS

## 18.4 Overall System Structure

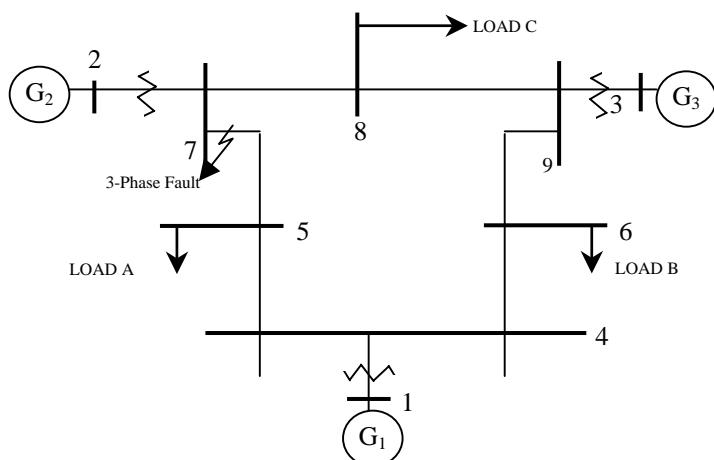
The schematic structure of the proposed MFLC is shown in Figure 18.5. It consists of: 1) the system dynamics used to evaluate the rotor angle of the machines; 2) a hierarchical fuzzy logic controller to control the plant; 3) the scaling factors to adjust the input/output values of the controller into proper ranges; 4) a limiter to constrain the control action within admissible limits. Typical input variables for the transient stability control are the rotor angle, angular speed, acceleration *etc.* A desired value was found for the rotor angle at post fault, hence the deviation of the rotor angle, the angular speed, the acceleration and the deviation of the shaft speed were all employed as inputs to the proposed controller. The positioning and realization of the dynamic breaking resistors is discussed in [2].



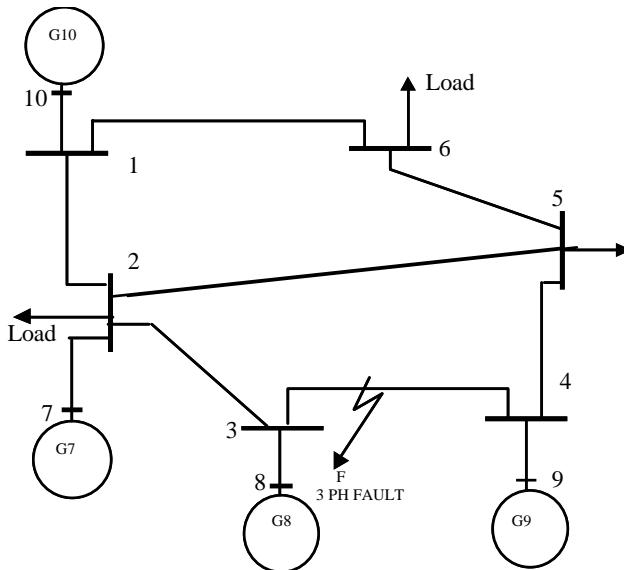
**Figure 18.5.** Overall schematic structure of the proposed HFLC

## 18.5 Implementation and Test Results

Two power networks are used to demonstrate the significance and applicability of the proposed controller: the Western States Coordinating Council (WSCC) test system and a four-generator test system [2, 3] are shown in Figures 18.6 and 18.7, respectively.



**Figure 18.6.** WSCC test system

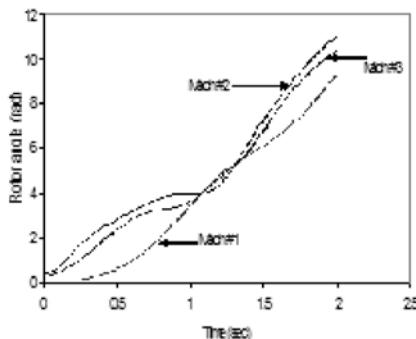


**Figure 18.7.** IEEE 4-generator test system

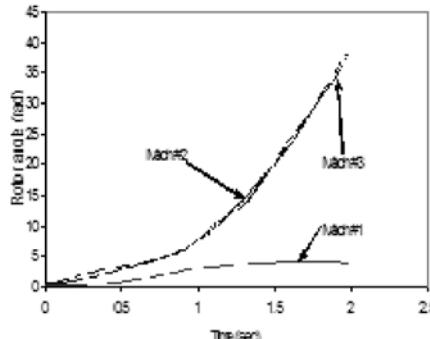
### 18.5.1 The WSCC Test System

The test system consists of 9-bus, 3 machines, 3 load system widely used in the literature and often referred to as the WSCC. A one-line impedance diagram for the system is given in Figure 18.6. Consider a three phase fault on the line between buses 5 and 7 near bus 7. Since the voltage at the fault point drops to zero, this results in the isolation of generator 2 from the system at bus 7. The fault is cleared by the opening of the circuit breaker at both ends of the line between 5 and 7. Under this “post fault” conditions, the voltage at bus 7 rises from zero, with the effect of having generator 2 reconnected to the system through the line between buses 7 and 8 as a weak link. For these conditions, a critical clearing time ( $t_c = 0.159$  s) was found.

Figure 18.8 shows the rotor angle oscillations of the three generators for a fault at  $t = 0$  and a clearing time of  $t = 0.159$  s. The rotor angles are shown to increase together and swing in unison. The curves span a period of 2 s. This is more than enough, since transient stability is decided during the first swing, if it exists. The results of another simulation for the same conditions and a clearing time of 0.160 s are shown in Figure 18.9. The curves clearly show the rotor angles two generators increasing relatively to the other without bound.



**Figure 18.8.** Stable fault cleared at 0.159 s



**Figure 18.9.** Unstable fault cleared at 0.160 s

Three fault durations as listed in Table 18.6 along with the fault clearing states are controlled. Table 18.7 shows the final desired states. The 0.160 s fault point on the fault trajectories is taken as the first initial state point. The trajectory of the error of the states during the simulation is shown in Figure 18.10 where the error reduces to less than 0.0001 in about 34 iterations. The resulting rotor angles *vs* time are shown in Figure 18.11. Note that all the swing curves are quite stable and remain near their respective desired states. The corresponding fuzzy controls *vs* time are shown in Figure 18.12.

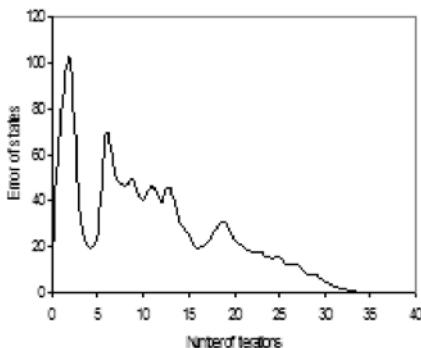
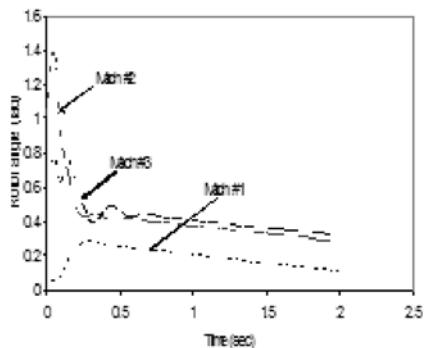
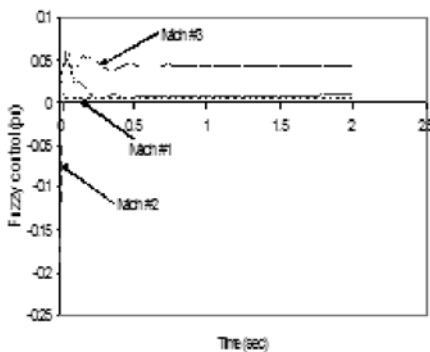
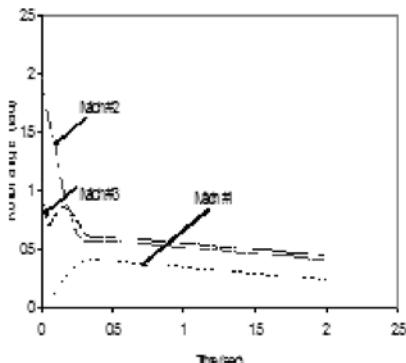
Figure 18.13 shows the rotor angles starting from a time of 0.266 s into the fault trajectory. The corresponding fuzzy controls are shown in Figure 18.14. In Figure 18.15 the rotor angles are shown starting from a time of 0.452 s into the fault trajectory with the corresponding fuzzy controls shown in Figure 18.16. Hence, with increase in the severity of the fault, the fuzzy control was able to damp gradually the oscillations of the system nicely.

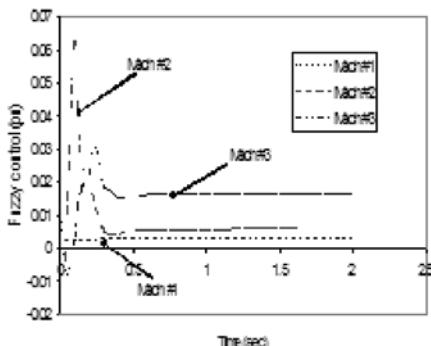
**Table 18.6.** Initial states for different fault durations

		Initial states values		
Rotor angle( $\delta$ )	Duration in seconds			
	0.160	0.266	0.452	
$\delta_1$	0.0487	0.0828	0.2610	
$\delta_2$	0.9983	2.1136	5.3733	
$\delta_3$	0.5925	1.0800	2.0127	
Rotor speed( $\omega$ )	Duration in seconds			
	0.160	0.266	0.452	
$\omega_1$	0.1541	0.4874	1.4089	
$\omega_2$	7.6622	12.7704	21.7097	
$\omega_3$	3.9071	4.9288	5.0043	

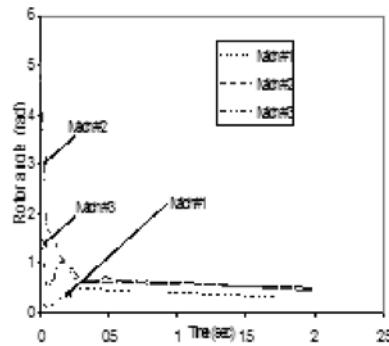
**Table 18.7.** Desired states

Desired state values					
$\delta_1$	$\delta_2$	$\delta_3$	$\omega_1$	$\omega_2$	$\omega_3$
0.03965	0.34437	0.22995	0.0	0.0	0.0

**Figure 18.10.** Error in the states *vs.* number of iterations for a fault clearing time of 0.16s**Figure 18.11.** Rotor angle trajectory for all machine for a fault clearing time of 0.16s**Figure 18.12.** Control trajectories for all machines for a fault clearing time of 0.16s**Figure 18.13.** Rotor angle trajectory for all machine for a fault clearing time of 0.266s



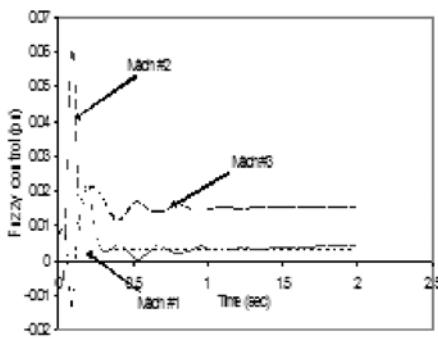
**Figure 18.14.** Control trajectories for all machines for a fault clearing time of 0.266s



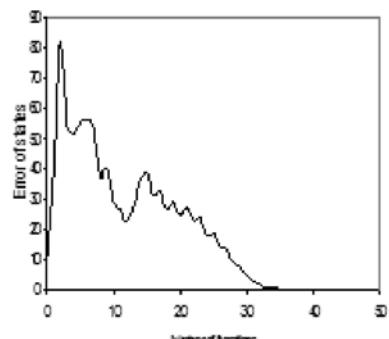
**Figure 18.15.** Rotor angle trajectory for all machine for a fault clearing time of 0.452s

### 18.5.2 Four-generator Test System

The system configuration is shown in Figure 18.7. All parameters used in this study can be found in [2]. A 3-phase to ground fault is considered on the line between buses 3 and 4, near bus 3. Since the voltage at the fault point drops to zero, this results in the isolation of generator 8 from the system at bus 3. This condition exhibits the “during-the-fault condition.” The fault is cleared by the opening of the circuit breakers at both ends of the line between 3 and 4. Under this post-fault condition, the voltage at bus 3 rises from zero, with the effect of having generator 8 reconnected to the system through the line between buses 3 and 2, thus leaving machine 8 with only a weak link to the rest of the system. A fault clearing time of approximately 0.4377s leads to an unstable post-fault trajectory for the uncontrolled system was found.

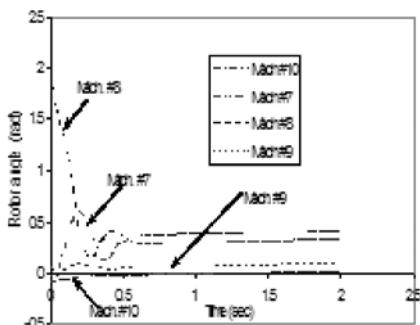


**Figure 18.16.** Control trajectories for all machines for a fault clearing time of 0.452s

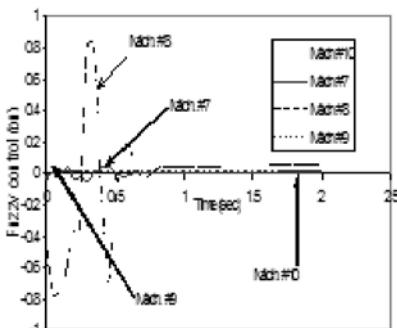


**Figure 18.17.** Error in the states vs number of iterations for a fault clearing time of 0.4377s

The fault durations along with the fault clearing states and the final desired state for this test system are the same as in [2] and are not repeated. The trajectory of the error of the states during the simulation is shown in Figure 18.17 where the error reduces to less than 0.0001 in about 36 iterations. With the 0.4377-s fault point on the fault trajectory taken as the first initial state point, the rotor angles *vs* time for this scenario is shown in Figure 18.18 and the corresponding fuzzy controls are shown in Figures 18.19 and 18.20. Note that the swing curves are stable and they all remain near their respective desired states.

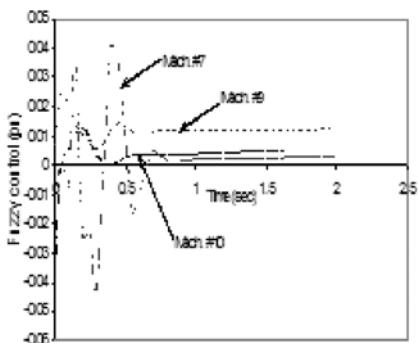


**Figure 18.18.** Rotor angle trajectory for all machine for a fault clearing time of 0.4377s

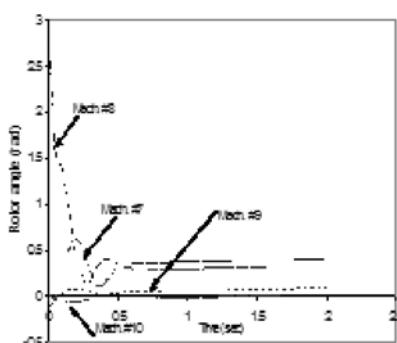


**Figure 18.19.** Control trajectories for all machines for a fault clearing time of 0.4377s

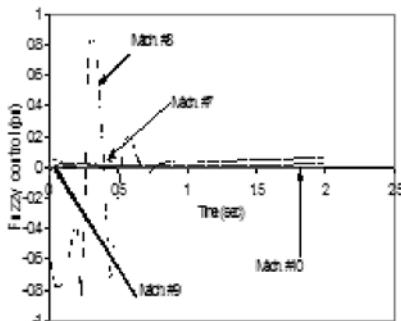
Figure 18.21 illustrate the rotor angle *vs* time from initial states of 0.5305 s into the fault. The corresponding fuzzy controls *vs* time are shown in Figures 18.22 and 18.23. Again, with the severity of the fault increased, the fuzzy control was able to gradually damp the oscillations of the system nicely and they also remain near their respective desired states.



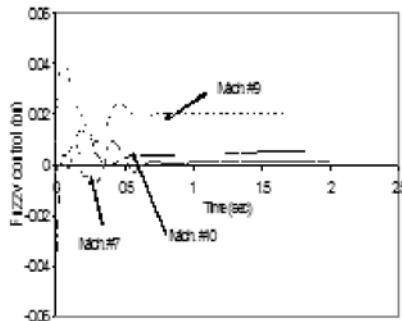
**Figure 18.20.** Control trajectories for Mach 7, 9, & 10 for a fault clearing time - 0.4377s



**Figure 18.21.** Rotor angle trajectory for all machine for a fault clearing time of 0.5305s



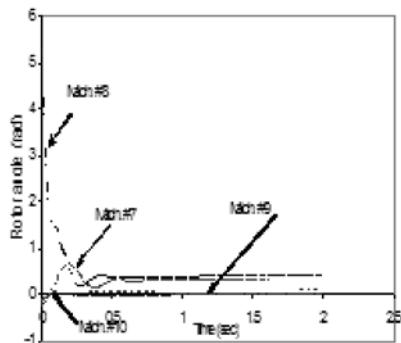
**Figure 18.22.** Control trajectories for all machines for a fault clearing time of  $0.5305\text{ s}$



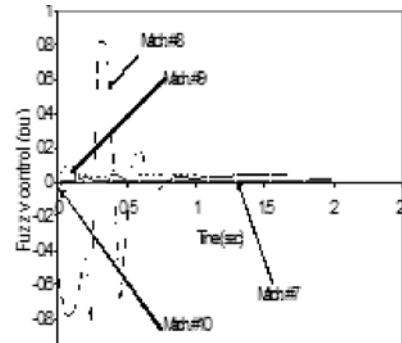
**Figure 18.23.** Control trajectories for Mach 7, 9, & 10 for a fault clearing time of  $0.5305\text{ s}$

Figures 18.24~18.26 illustrate similar results even when starting at  $0.79578\text{ s}$  into the fault. The well-damped swing curves and the almost smooth control effort indicate the effectiveness of the proposed control strategy for the enhancement of transient stability for the multi-machine power system. Note that the final desired states are the same as the previous cases. In spite of the increased severity of this fault condition, the swing curves are well damped and remain near their respective desired states values.

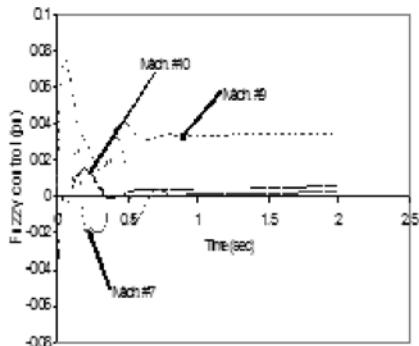
A comparison of the MLFC with the two-level hierarchy controllers in [2] was made. Plots of the rotor angles for machine 8 are shown in Figures 27~29 for fault



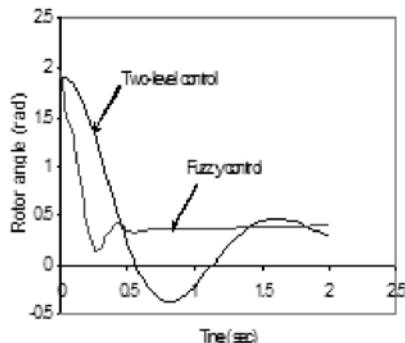
**Figure 18.24.** Rotor angle trajectory for all machine for a fault clearing time of  $0.7957\text{ s}$



**Figure 18.25.** Control trajectories for all machines for a fault clearing time of  $0.7957\text{ s}$



**Figure 18.26.** Control trajectories for Mach 7, 9, & 10 for a fault clearing time - 0.7957s

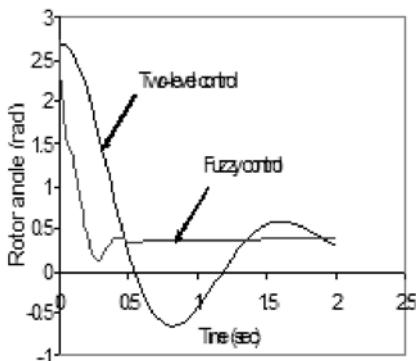


**Figure 18.27.** Rotor angle trajectory for comparison for a fault clearing time of 0.4377s

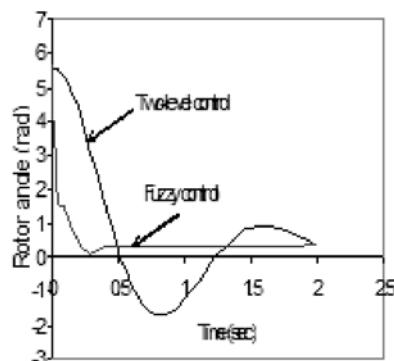
clearing time of 0.4347 s, 0.5350 s, and 0.7958 s respectively. It is seen that the fuzzy logic controller was able to damp the initial rotor angle smoothly with less oscillations compared to the two-level hierarchy controllers.

## 18.6 Conclusion

A multilayer fuzzy logic control scheme has been proposed in this chapter to improve the transient stability of power systems. During a disturbance, the generator speed deviation, its derivative, rotor angle deviation from the desired, and the rate of change of the rotor angle are computed from real time measurements and presented as inputs to the MLFC. The first layer, termed the



**Figure 18.28.** Rotor angle trajectory for comparison for a fault clearing time of 0.5305s



**Figure 18.29.** Rotor angle trajectory for comparison for a fault clearing time of 0.7957s

supervisory layer specifies the region of operation of the SCs within the second layer, called the execution layer. The outputs of the SCs are then fuzzily combined to achieve the overall objective of the system. The simulation results show that the proposed fuzzy logic controller can effectively enhance the stability of a power system.

## References

- [1] M. Dicorato, M. L. Scala, and P. Scarpellini, "A Corrective control for angle and voltage stability enhancement on the transient time-scale," *IEEE Trans. Power Syst.*, Vol. 15, pp 1345–1353, Nov. 2000.
- [2] A. Rubaai and A. R. Ofoli, "Design and analysis of nonlinear digital controllers-based two-level hierarchy for electric utility industry," *IEEE Trans. Ind. Applicat.*, Vol. 39, pp 395–407, Mar./Apr. 2003.
- [3] K. M. Son and J. K. Park, "On the robust LQG control of TCSC for damping power system oscillations," *IEEE Trans. Power Syst.*, Vol. 15, pp 1306–1312, Nov. 2000.
- [4] Z. Xi, D. Cheng, Q. Lu, and S. Mei, "Nonlinear decentralized controller design for multi-machine power systems using Hamiltonian function method," *Automatica*, Vol. 38, No. 3, pp 527–534, 2002.
- [5] Z. Xi, D. Cheng, and Q. Lu, "Nonlinear decentralized saturated controller design for power systems," *IEEE Trans. Contr. Syst. Technol.*, Vol. 11, pp 539–547, July 2003.
- [6] P. K. Dash, S. Mishra, and G. Panda, "Damping multi-model power system oscillation using a hybrid fuzzy controller for series connected facts devices," *IEEE Trans. Power Syst.*, Vol. 15, pp 1360–1366, Nov. 2000.
- [7] R. B. Chedid, S. H. Karaki, and C. El-Chamali, "Adaptive fuzzy control for wind-diesel weak power systems," *IEEE Trans. Energy Conversion*, Vol. 15, pp 71–78, Mar. 2000.
- [8] Y.-K. Chen, C.-H. Yang, and Y.-C. Wu, "Robust fuzzy controlled photovoltaic power inverter with Taguchi method," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. 38, pp 940–954, July 2002.
- [9] T. Hiyama, M. Mishiro, and H. Kihara, "Fuzzy logic switching of thyristor controlled braking resistor considering coordination with SVC," *IEEE Trans. Power Delivery*, Vol. 10, pp 2020–2026, Oct. 1995.
- [10] O. Cordon and F. Herrera, "Linguistic modeling Hierarchical systems of Linguistic rules," *IEEE Trans. Fuzzy Syst.*, Vol. 10, pp 2–20, Feb. 2002.
- [11] R. R. Yager, "On the construction of hierarchical fuzzy systems models," *IEEE Trans. Syst., Man, Cybern.*, Vol. 28, pp 55–66, Feb. 1998.
- [12] ----, "On a Hierarchical structure of fuzzy modeling and control," *IEEE Trans. Syst., Man, Cybern.*, Vol. 23, pp 1189–1197, Aug. 1993.
- [13] A. E. Gegov and P. M. Frank, "Hierarchical fuzzy control of multivariable systems," *Fuzzy Sets Syst.*, Vol. 72, pp 299–310, 1995.
- [14] C. V. S. Raju and J. Zhou, "Adaptive Hierarchical fuzzy controller," *IEEE Trans. Syst., Man, Cybern.*, Vol. 23, pp 973–980, Aug. 1993.
- [15] E. H. Mamdani, "Twenty years of fuzzy control: Experiences gained and lessons learnt," in *Proc. 2nd IEEE Int. Conf. Fuzzy Systems*, Vol. 1, 1993, pp 339–344.

---

## Fuzzy Predictive Control for Power Plants

Aldo Z. Cipriano

### 19.1 Introduction

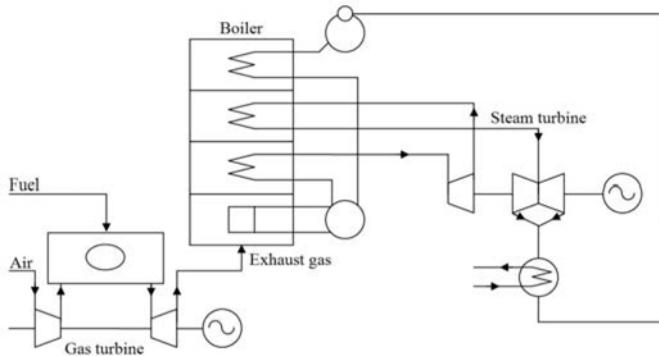
The power generation industry has experienced significant changes in recent decades [1]. Deregulation has brought about a much greater emphasis on economic and financial factors, while growing concerns about the environment have led to restrictions on gaseous emissions and a more intensive search for renewable energy sources. Technological progress has also given rise to major developments in the way of cheaper and more efficient solutions. Whereas early power plant boilers were fired by steam coal or coal gas, operated at 0.9 MPa and 150°C and turned 30kW generators, today's plants use advanced boilers with new metal alloys and turbines, operate under supercritical conditions (28.5 MPa and 600 °C) and generate 1,300 MW. Modern combined-cycle gas turbines with a heat recovery steam generator can reach efficiencies of 50~60% [2]. Generating plants now also have advanced instrumentation and complex computerized monitoring, supervisory and control systems (Supervisory Control and Data Acquisition, SCADA, or Distributed Control System, DCS). In the wake of these technological advances, new and greater demands on the control systems have emerged, stimulating the development of new control strategies and algorithms.

Figure 19.1 shows a highly simplified modern combined-cycle power plant that uses a gas turbine and a steam turbine to generate electricity [3]. The two turbines are combined into a single cycle and energy is transferred from one to the other. The exhaust gases from the gas turbine are used to provide heat to the boiler for producing steam, which in turn is fed to the steam turbine.

The control system of a modern power plant has an interacting structure and a hierarchy of control loops consisting of up to three levels: 1) the unit control level, where the overall unit objective of meeting the power system demand is transformed into more specific control tasks; 2) the principal regulation level, where the main process variables are controlled by an appropriate combination of feed forward and feedback actions; and 3) the dependent loop level, where the physical devices for modulating basic process variables are controlled in a substantially independent manner [4].

The main components of a plant control system are the boiler and turbine control subsystems [5]. The principal set-point is the steam demand signal, also known as the master control signal, from which the set-points for other control loops are derived. The variables that are usually controlled are the electric load, the

evaporation pressure to keep the thermal energy stores in the evaporator within acceptable margins, the temperature at the hottest zones of the plant, and the mass of water in the evaporator.



**Figure 19.1.** Combined-cycle power plant

The specific behavior of the plant is determined by the control configuration [2]. One of the most common configurations is known as boiler following mode or constant pressure mode, which uses the main steam governor as a fast-acting load controller. Opening the governor valves and releasing the stored energy in the boiler reduces the generated output. A more efficient alternative configuration is the turbine following mode, in which the governor valves remain fully open and the position of the main steam governor valve is controlled by the valve outlet pressure. In this mode the generator units operate at maximum capacity but are unable to respond to frequency deviations. A third control configuration is sliding pressure control, in which vapor pressure is variable thus enabling faster unit loading and operation of the turbines at lower temperatures and pressures.

In terms of process control, a power plant may be considered as an interacting multi-input multi-output process that is non-linear and time-varying, with disturbances and significant changes in its set-points due to variations in demand. With such characteristics it is impossible to ensure a satisfactory dynamic performance using only PID controllers, and numerous supervisory control strategies have been designed as a result. These strategies maintain PID controllers in the final loops but incorporate more advanced techniques at the higher levels, such as multivariable control, LQG, self-tuning adaptive control, robust control, predictive control, fuzzy control, neural networks and genetic algorithms [5, 6]. Two of these – model predictive control and fuzzy control – will be analyzed in this study.

## 19.2 Model Predictive Control

Model predictive control (MPC) involves a complete family of controllers [7] whose basic concepts are: 1) use of an explicit model to predict process outputs at

discrete future time instants over a prediction horizon; 2) computation of a sequence of future control actions through the optimization of an objective function with given operating constraints and desired reference trajectories for process outputs; and 3) repetition of the optimization process at each sampling instant and application of the first value of the calculated control sequence (receding horizon strategy). These three characteristics allow MPC to handle multivariable, non-minimum phase, open-loop unstable and non-linear processes with a long time delay and the inclusion, if necessary, of constraints for manipulated and/or controlled variables.

As shown in Figure 19.2, at instant  $t$  future process outputs are predicted over the prediction horizon  $N_p$  using an explicit model. The predictions depend on known values of the manipulated and controlled variables and future control actions  $u(t+j | t)$  over the control horizon  $N_u$ . It is assumed that  $N_u \leq N_p$  and that  $u(t+j | t)$  remains constant for  $j = N_u, \dots, N_p$ . The prediction horizon  $N_p$  is related to the response time of the process and  $N_u$  is usually chosen so as to be equal to the order of the model.

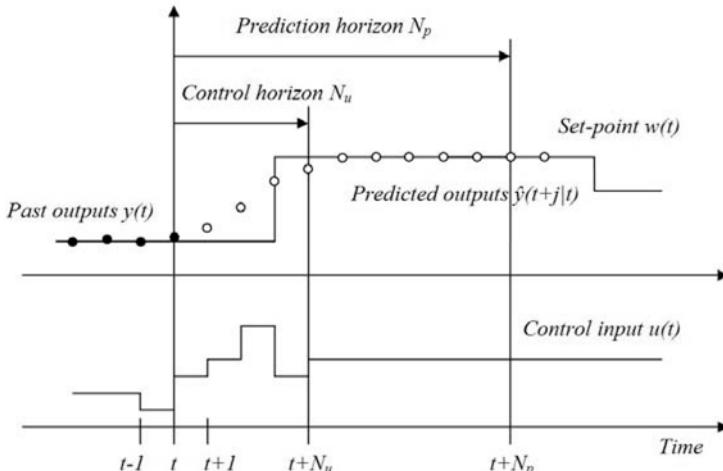


Figure 19.2. MPC strategy

The control sequence is obtained through the optimization of an objective function that describes the goals the control strategy is intended to achieve. The optimization process may involve hard or soft constraints. In classical MPC, an objective function minimizes both the error between predicted outputs and the set-points during the prediction horizon and the control effort  $\Delta u$  during the control horizon. The function may be expressed as

$$J = \sum_{j=N_l}^{N_p} \delta(j) (\hat{y}(t+j | t) - w(t+j))^2 + \sum_{j=1}^{N_u} \lambda(j) (\Delta u(t+j-1 | t))^2 \quad (19.1)$$

where  $\hat{y}(t+j | t)$  is the expected value of the predicted output at instant  $t+j$  with known history up to time  $t$ . Parameters  $\delta(j)$  and  $\lambda(j)$ , which weight each term in the optimization problem, can change while time is evolving to prioritize the final error or the transient period, among other things. Parameter  $N_l$  is usually set as the control action's dead time. For linear unconstrained systems this optimization problem is tractable and convex and can be solved analytically, but in general applications it is common to take into account constraints or non-linearities in the process, and in such cases the optimization problem must be solved using iterative numerical methods.

Numerous alternative versions of MPC algorithms have been developed based on the foregoing concepts. They differ from one other as regards the model used to represent the process and disturbances, the cost function to be minimized, the constraints applied to each variable, or the optimization algorithm employed. Among these versions are [7] model predictive heuristic control (MPHC), dynamic matrix control (DMC), extended horizon adaptive control (EHAC), extended prediction self adaptive control (EPSAC), generalized predictive control (GPC), multistep multivariable adaptive control (MUSMAR), multipredictor receding horizon adaptive control (MURHAC), predictive functional control (PFC) and unified predictive control (UPC).

MPC has achieved widespread popularity in process industries, where it is currently employed in more applications than any other advanced control technique. A brief list of companies offering predictive control products and the names under which they are marketed includes Adersa (HIECON, PFC), Aspen Tech (DMC-plus, Aspen Target), Continental Controls Inc. (MVC), DOT Products (NOVA-NLC), Honeywell Hi-Spec (RMPCT), Pavilion Technologies (Process Perfecter), Shell Global Solutions (SMOC-II), and Invensys (Connoisseur) [8].

MPC technology has advanced significantly over the last 25 years, stimulating a parallel growth in the quantity of industrial applications. A study by Qin and Badgwell [8] found that the number of such applications doubled between 1995 and 1999, with vendors reporting a total of 4635 for the latter year. However, only a small number of these (93) were designed using non-linear models, even though the great majority of processes are non-linear. If a non-linear model is employed, and if, in addition, the objective function is not quadratic and/or subject to constraints, the optimization problem will not be convex, difficulties will arise regarding the existence of local optima and computational complexity will be much greater. These drawbacks limit the application of such models to slow processes.

Among the ten MPC products listed above, six of them (HIECON, PFC, DMC-plus, RMPCT, SMOC-II and Connoisseur) use linear models, while the remaining four (Aspen Target, MVC, NOVA-NLC and Process Perfecter) employ non-linear ones. These latter models differ primarily as regards the structure of the process model, but also in their objective functions, constraints, set-point trajectories, horizons and methods of deriving the optimization problem solution.

### 19.3 Model Predictive Control for Power Plants

Much work has been done in the development of MPC for power plants, and the literature on the topic is abundant. Some of the relevant publications are described below.

Amin *et al.* [9] present a multivariable self-tuning controller for controlling the superheat steam temperature, reheat steam temperature, and superheat steam pressure of a 600-MW oil-fired power plant boiler with a Combustion Engineering steam generator and a Westinghouse turbine-generator. Their model for approximating the relationship between outputs and input at constant load level is a linear ARMAX model whose parameters are determined using recursive least-squares estimation.

Hogg and El-Rabaie [10] propose a multivariable self-tuning generalized predictive controller for a boiler system simulator. To characterize the system they use a Cromby model, and set out a mathematical description of a 200-MW oil-fired power plant consisting of 14 non-linear first-order differential equations with 82 additional algebraic equations, both linear and non-linear. Two configurations, the boiler following mode and the turbine following mode, are contemplated, and the controlled variables are superheated steam pressure, superheated steam temperature and reheated steam pressure. Drum water volume is controlled by means of a PI controller. Finally, their design uses linear time-series models.

Pérez *et al.* [11] apply the adaptive predictive control system (APCS) to the control and optimization of the Pasajes de San Juan power station in Spain. APCS uses an adaptive predictive model, a driver block, and an adaptation mechanism that minimizes the prediction error and enables the driver block to redesign the desired output trajectory for optimizing control system performance.

Prasad *et al.* [12] describe an application developed for a simulator similar to the one in [10] for a 200-MW oil-fired unit at Ballylumford Power Station in Northern Ireland. In this case an NMPC (non-linear model predictive control) structure is employed, but with the same state equation for predicting dynamic system behavior. An extended Kalman filter is utilized for state and parameter estimation.

Kruger *et al.* [13] apply predictive control to the design of a start-up strategy for a high-pressure bypass and a drum-type boiler consisting of furnace, economizer, evaporator, superheaters and attemperator. The controller design is based on a non-linear process model, and a cost function was formulated that includes fuel consumption and boiler start-up time, with hard constraints for the thermal stress on thick-walled components. The optimization provides the start-up phase values, which include the controlled variables (steam pressure, steam temperature and steam mass flow rate) plus the control inputs (fuel flow rate and high-pressure bypass control valve positions). These optimal variables are calculated using sequential quadratic programming (SQP) techniques.

In Peng *et al.* [14] a modeling and control method is proposed for thermal power plants with non-linear dynamics that vary with load. A load-dependent exponential ARX model, identified off-line, is used to design a constrained long-range multivariable multistep predictive control strategy that is evaluated by simulation of a 600-MW thermal power plant. The exponential ARX model very

closely describes the nonlinear dynamic behavior of the plant over the entire operating range. This long-range predictive controller displays much better control performance than either gain-scheduling PID or GPC using a global linear ARX model. A later study [15] proposes the use of RBF-ARX models, which are combinations of linear ARX models and Gaussian radial basis function (RBF) neural networks, to design more robust MPC controllers. The stability of these controllers is ensured by using a time-varying parameter-dependent Lyapunov function and Linear Matrix Inequalities.

Poncia [5] describes the application of MPC to a 320-MW oiled-fired plant. A non-linear dynamic simulator was developed from first principles equations. The simulator was validated with data from the Castel S. Giovanni plant owned by Enel S.p.A. The controlled variables are the temperature at superheater outlet and the evaporation pressure, while the manipulated variables are the water and fuel mass flows. Electrical power demand is used for compensation purposes. A linear low-complexity state-space model of the plant was obtained using the subspace model identification method. The control algorithm is the solution of an optimization problem with a quadratic objective function and finite final time.

In [16] a predictive supervisory controller design is presented that optimizes an objective function in order to determine the set-points for a given regulatory level. The proposed controller is applied to the gas turbine of a thermal power plant and is compared to a control strategy in which the optimal set-points are constant. The controller provides a quick way of implementing the predictive control algorithm without numerical optimization. The benefits obtained for the gas turbine ranged from 2.4 to 4.4%.

Finally, we also note that the study [8] contains 21 applications of predictive control in utilities developed by Aspen Technology, Continentals Controls, Honeywell Hi-Spec and Invensys.

## 19.4 Fuzzy Control of Power Plants

As has occurred in other industries, the electrical power industry has been the target of a large quantity of fuzzy logic applications. Bansal [17], for example, enumerates 282 publications on fuzzy set theory applications to power systems in 1994~2001 alone, with other researchers adding more to that total for the same period. Though a similar figure has not been counted for power plants, developed applications in this area are also numerous and embrace a wide variety of situations. A few of these are described in what follows.

In [18], a fuzzy expert system is described that supports operators of fossil power plants in creating optimal start-up schedules and executing them accurately. In [19], a fuzzy-logic-based supervisor monitors the overall plant and handles the tasks of coordination, fault diagnosis, fault isolation, and fault accommodation. Holmes and Ray [20] describe a fuzzy damage-mitigating control of a fossil-fueled power plant with a two-tier structure, the lower tier a linear sampled-data controller that traces a reference trajectory vector and the upper tier a fuzzy-logic-based damage controller which trades off the system's dynamic performance with structural damage in critical components.

Garduno-Ramírez and Lee [21] present a two-level hierarchical control scheme for a wide-range of fossil power plants. It consists of a fuzzy reference governor at the supervisory level and a feedforward-feedback control strategy with a MIMO fuzzy inference system and PID controllers in a multiloop configuration at the control level; the fuzzy reference governor generates the set-point trajectory to command the unit along any load demand pattern according to a variable pressure operating policy. A fuzzy-logic-based methodology of life-extending control for robust wide-range operation of fossil power plants including load-following and hot start-up is described in [22]. Si Fodil *et al.* [23] present a real-time fuzzy controller for power axial offset and control rod insertion in a pressurized water nuclear reactor developed by Electricité de France. Jianghong *et al.* [24] propose a multivariable fuzzy PID control system which they have put into operation in a 200-MW thermal unit at the Yangzhou electrical power plant in JangSu, China. Finally, Wang *et al.* [25] develop a hybrid classical/fuzzy control methodology to integrate low-level machine control and high-level supervision for the steam temperature and water level processes of a power plant boiler.

Depending on the particular structure of the consequent proposition, the fuzzy control systems may be classified [26] as follows: 1) rule-based linguistic fuzzy systems, if the antecedent and consequent are fuzzy propositions; 2) fuzzy relational systems, a generalization of the linguistic model allowing one particular antecedent proposition to be associated with several different consequent propositions *via* a fuzzy relation; and 3) Takagi-Sugeno fuzzy systems, if the consequent is a crisp function of the antecedent variables.

Although it is true that in the majority of existing applications, the solutions using fuzzy logic exhibit better performance than those employing more conventional techniques, some evaluations comparing fuzzy control and model-based advanced control of power plants have concluded that the latter offers a better cost-benefit relationship. Moelbak and Mortensen [27], for example, studied the control of a Skaerbaekvaerter unit, a 265-MW coal-fired unit equipped with a Benson boiler. The plant superheater was experiencing control problems due to oscillations in certain variables that constituted the limiting factor in the load-following capability of the boiler and hence the unit as a whole. A better control of these variables was therefore necessary to improve the load-following capability of the unit.

With this in mind, Moelbak and Mortensen compared the following controllers: PI control; fuzzy PI control complemented by a set of “breaking rules” to handle the high-order dynamics; a simple model-based strategy using cascade control with a PI controller in the inner loop, a P-controller in the outer loop and a simple model-based compensation function for the high-order and load dependent dynamics of the superheater (PTx control); and generalized predictive control with on-line identification of the model parameters.

Table 19.1 summarizes the results of Moelbak and Mortensen’s comparison. They conclude that, since the thermal processes are quite well known in mathematical terms, modeling of the processes should be direct. This favors methods like LQG, GPC and PTx, as opposed to fuzzy-based methods which are based on indirect modeling. From this conclusion we may further deduce that, to obtain better performance from a power plant control system, it may be useful to

integrate different control methodologies within one unified approach – predictive control with fuzzy systems, for example – and seek to maximize the contribution of each to the global benefits.

**Table 19.1.** Comparison of different superheater controllers

Controller type	Control performance	Resource demand
PI	Low	Low
Fuzzy	Medium	High
Simple model-based	High	Medium
Adaptive GPC	High	High

## 19.5 Fuzzy Systems and Model Predictive Control

The possibility of integrating fuzzy systems with optimization, the latter including predictive control, arose in one of the first studies on decision-making in a fuzzy environment [28]. Decision making in a fuzzy environment means a decision process in which the goals and/or the constraints, but not necessary the system being controlled, are fuzzy in nature. Fuzzy goals and fuzzy constraints can be defined precisely as fuzzy sets in the space of alternatives. A fuzzy decision may then be viewed as an intersection of the given goals and constraints, and a maximizing decision is defined as a point in the space of alternatives at which the membership function of a fuzzy decision attains its maximum value [28].

Another way of integrating fuzzy systems and optimization consists in solving the optimization problem on the assumption that the system – or more precisely, the model of the system – is a dynamic fuzzy one. The most widely used formulation for this purpose is an extension of the Takagi-Sugeno (TS) fuzzy system. If the consequents are linear ARX expressions, a nonlinear dynamic fuzzy system is described by the following set of  $K$  TS rules [26]:

$$\begin{aligned}
 R_i : & \text{ IF } y(t) \text{ is } A_{i1} \text{ and } y(t-1) \text{ is } A_{i2} \text{ and } \dots y(t-n_y+1) \text{ is } A_{in_y} \\
 & \text{ and } u(t) \text{ is } B_{i1} \text{ and } u(t-1) \text{ is } B_{i2} \text{ and } \dots u(t-n_u+1) \text{ is } B_{in_u}
 \end{aligned} \tag{19.2}$$

$$\text{THEN } y_i(t+1) = \sum_{j=1}^{n_y} a_{ij} y(t-j+1) + \sum_{j=1}^{n_u} b_{ij} u(t-j+1)$$

where  $A_{ij}$  and  $B_{ij}$  are fuzzy sets defined by multivariate membership functions. In the Takagi-Sugeno model, the inference is reduced to a simple algebraic expression similar to the fuzzy-mean defuzzification algorithm [29].

A nonlinear fuzzy model in state-space can also be formulated using the set of

$K$  rules, as follows:

$$R_i : \text{IF } x(t) \text{ is } A_i \text{ and } u(t) \text{ is } B_i \text{ THEN } x(t+1) = M_i x(t) + N_i u(t) \quad (19.3)$$

## 19.6 Integrating Fuzzy Systems and Model Prediction for Power Plant Control

Various applications for power plants have been developed and published that integrate fuzzy systems and predictive control using the methodologies of Bellman and Zadeh [28] and Takagi and Sugeno [29]. A few of these are described briefly below, followed by an examination of two applications in greater detail. The first of the two (Section 19.7) uses the TS model to develop a non-linear predictive controller for a boiler, while the second (Section 19.8) involves a predictive controller for a solar plant whose objective function and constraints are fuzzy.

One of the earliest applications integrating fuzzy systems and model prediction was developed by General Electric, and consists of a novel hierarchical coordination control for the cycling of steam turbines [30]. The scheme includes a projected load demand profile, a model predictive control algorithm, and a higher-level fuzzy logic knowledge base. The load demand profile provides a desired trajectory for plant megawatt output. The model predictive control algorithm determines the optimal actuator commands to track the desired load profile and minimizes the cost of power generation. Hard system constraints such as maximum safe operating pressure or valve closure rate are explicitly included in the algorithm, as are soft constraints like desired pressure at the end of a load change. The fuzzy logic knowledge base provides long-term optimization guidance by selecting present operating priorities for the soft constraints based on present and future demand. At the beginning of each stage in the load profile, the knowledge base provides the MPC with a set of weights for its cost function that govern how it executes the cycling stage. The controller displays good load tracking and long-term operating cost minimization characteristics, and performs better than a standard MPC with fixed weighting matrices.

Huaguang and Bien [31] and Huaguang and Cai [32] describe a fuzzy system identification method and a multivariable fuzzy generalized predictive controller that they applied to a simulated 200-MW boiler-turbine unit. The manipulated variables are the governor valve position and the fuel-to-air ratio, while the controlled variables are power output and steam pressure. The identification method was shown to be very accurate. The simulation demonstrated that the fuzzy GPC has a higher load change tracking speed, a steadier dynamic response to pressure and a lower sensitivity to external disturbances than the conventional boiler following load control system.

One of the main reasons for employing the Takagi-Sugeno fuzzy methodology is that the dynamics of a non-linear system are easily obtained by linearization of non-linear models near different operating points. This characteristic has been put to use in [33] to design a control system for the internal temperature of the

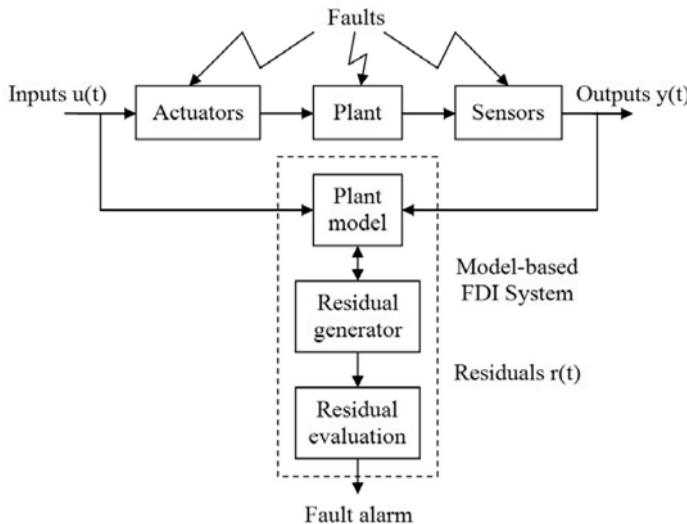
downcomers during the steam generator start-up process. Four fuzzy sets were defined, and for each of these four regions a state-space linear model was identified. Once these local models were obtained, a global fuzzy systems dynamic model was implemented. The corresponding optimization problems were solved using the MATLAB® Predictive Control Toolbox.

A similar scheme is proposed in [34], in this case for the design of a steam generator water level controller in the secondary circuit of a nuclear plant to prevent reactor shutdown, a common occurrence in plants that exhibit strong non-minimum phase characteristics. The same technique is used in [35] for applying predictive functional control to the control of vapor temperature in thermal plants. There, the linear models of each rule consequent are finite impulse responses (FIR) and the controller is a cascade superheated steam temperature control system incorporating the compensation effect for a measured disturbance.

Gardano-Ramírez and Lee [36] have developed a variation on this method using a neural representation of a fuzzy controller, which they apply to enhance the load-following capabilities of a fossil fuel power unit subject to a given power-pressure operating policy. To achieve a better distribution of tasks, the authors employ a feedforward/feedback control scheme. The feedforward control improves the maneuverability of the power unit throughout the range of operation, while the feedback compensates for uncertainties and unknown disturbances around the command trajectories. The feedforward controller was designed as a set of MISO fuzzy inference systems that approximate the inverse steady-state of the power unit across the entire operating range. They are implemented as TS fuzzy systems, which may be represented by a forward neural network. The ability of the proposed knowledge-based controller to enhance effectively the load-following capabilities of a fossil fuel power unit was demonstrated through simulation experiments.

One of the most significant applications of predictive models is in the design of fault detection and isolation systems (FDI). A fault may be defined as an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable, usual or standard condition [37]. Fault detection refers to the determination of the faults present in the system and the time of detection, while fault isolation, which follows fault detection, is the determination of the kind, location and time of detection of a fault. As is illustrated in Figure 19.3, the residuals are the fault indicator variables in a model-based FDI system and are given as the deviation between measurements and model-equation-based computations. By integrating fault detection and isolation algorithms with control algorithms, a fault tolerant control (FTC) system can be structured [38].

Simani *et al.* [37] report 46 published applications of model-based fault detection in power plants and thermal processes, and 36 such applications for combustion engines and turbines. In one of these studies the same authors propose the use of fuzzy models to design an FDI system for an industrial turbine in a 120-MW power plant at Pont-sur-Sambre in France. The consequents are MISO submodels of state equations identified from time-series data supplied by sensors for the following variables: gas flow, turbine valves opening, superheater spray flow, gas dampers, air flow, steam pressure, main stream temperature, and reheat stream temperature.



**Figure 19.3.** Scheme for a model-based fault detection system

The study looked at four faults: compressor contamination (system fault), turbine damage (system fault), thermocouple sensor fault (output sensor fault), and controller actuator faulty (actuator fault). The results demonstrated that fuzzy models identified using the MATLAB® Fuzzy Modeling and Identification Toolbox and the Gustafson-Kessel clustering algorithm yield indicators for fault detection and isolation that are superior to the classical observers [37].

## 19.7 Supervisory Fuzzy Predictive Control of a Boiler

Sáez and Cipriano [39] compare by simulation two optimal supervisory control designs for a boiler. The first is based on a linear model and the second on a dynamic fuzzy model. The regulatory control includes PI controllers for the superheated steam pressure  $p_s$ , the drum water level  $L$ , the furnace gas pressure  $p_g$  and the superheated steam temperature  $T_s$ . The manipulated variables are fuel flow  $w_f$ , feedwater flow  $w_e$ , air flow to the furnace  $w_A$  and attemperator water flow  $w_a$ . The supervisory level provides the set-points  $p_s'$  and  $L'$  for the controlled variables  $p_s$  and  $L$ ; the set-points  $p_g'$  and  $T_s'$  remain constant.

The objective function incorporates both the economic and the regulatory level objectives. The economic objective is to minimize the operating costs  $J_{Cf}$ . With fuel flow  $w_f$  and feedwater flow  $w_e$  as the main process inputs, the economic objective function  $J_{Cf}$  is given by

$$J_{Cf} = \sum_{j=1}^N C_f w_f(t+j-1) + \sum_{j=1}^N C_e w_e(t+j-1) + CF \quad (19.4)$$

where  $C_f$  is the fuel costs,  $C_e$  the feedwater supply cost,  $CF$  the fixed operating costs, and  $N$  the prediction horizon.

The regulatory objective is to minimize  $J_{Cr}$ , the weighted sum of the set-point trajectory following an error and the control action effort. The objective function to be optimized at the supervisory level is thus given by  $J = J_{Cf} + \eta J_{Cr}$ , where  $\eta$  is the weighting factor.

Also important in this problem are the operating constraints such as the lower limit of drum water level for boiler safety ( $L(t) \geq L_{\min}$ ), the lower limits of steam flow and fuel flow due to pump operating limits ( $w_e(t) \geq w_{e\min}$  and  $w_f(t) \geq w_{f\min}$ ), and the upper limits of steam flow and fuel flow due to pipe capacity limitations ( $w_e(t) \leq w_{e\max}$  and  $w_f(t) \leq w_{f\max}$ ).

For the design of the linear model-based supervisory predictive controller, CARIMA models were used for superheated steam pressure and drum water level. The order and parameters of both models are obtained using a regression method. Solving the optimization problem required that PI controller models also be incorporated. As for the non-linear model-based supervisory predictive controller, the design employed a TS fuzzy model with two rules for each variable. Both the fuzzy membership functions and the consequent submodel parameters were determined using the method described in [39].

The optimization solution in both cases was found by applying the Kuhn-Tucker conditions and using the MATLAB® Optimization Toolbox to solve the resulting system of equations numerically.

The two proposed supervisory predictive controllers, based on linear (LMPC) and non-linear (FMPC) models respectively, are then compared with a control strategy in which the optimal set-points are constant (CSP). These constant optimal set-points are calculated by static optimization of the objective function defined in Equation (19.4) using a static model of the boiler. In Table 19.2, the mean values of the objective functions  $J_{Cf}$ ,  $J_{Cr}$  and  $J = J_{Cf} + J_{Cr}$  are derived using the simulation results.

According to these figures, operating costs decrease when supervisory predictive control is used. The savings under linear control range from 1.79% to 5.80% when the coefficient  $\eta$  changes between 1 and 0.5. An additional saving of 0.48% to 0.37% is added to the economic performance function when the non-linear fuzzy model is employed. These figures could be further improved if a multi-step fuzzy method were utilized for prediction, but this would require greater computational effort.

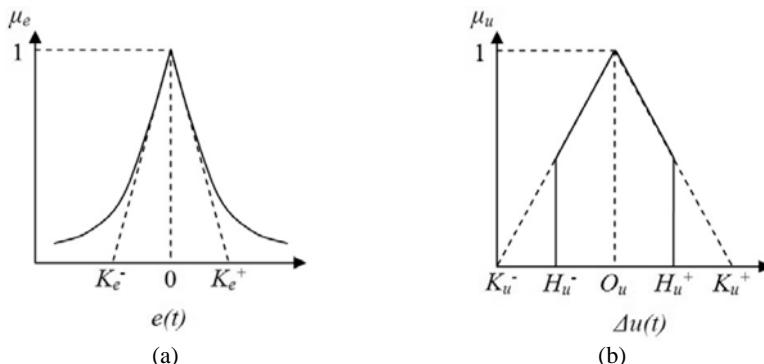
**Table 19.2.** Mean values of the objective functions for (a)  $\eta = 1.0$  , (b)  $\eta = 0.5$ 

	(a) $J_{Cf}[\$\$]$	(a) $J_{Cr}[\$\$]$	(a) $J[\$\$]$	(b) $J_{Cf}[\$\$]$	(b) $J_{Cr}[\$\$]$	(b) $J[\$\$]$
CSP	135,806	38,161	173,967	128,532	42,418	149,741
LMPC	133,378	24,113	157,491	121,075	37,450	139,800
FMPC	132,729	21,032	153,761	120,600	33,981	137,591

## 19.8 Predictive Control of a Solar Power Plant with Fuzzy Goals and Constraints

In classical MPC, the objective function is generally quadratic and its parameters are determined by trial and error. Because the objective function generated under this method is only an approximation of the desired control performance, we propose a new methodology that uses fuzzy goals and constraints instead of the typical quadratic criterion, thus affording greater flexibility in the definition of the optimization criteria. The result is that both the fuzzy membership function for the goals or constraints and their aggregation operator can be arbitrarily selected [40].

The satisfaction of the goals or constraints present in fuzzy predictive control can be represented with a fuzzy membership function as shown in Figure 19.4. The objective is to minimize the error  $e(t)$  and keep the change in the control action  $\Delta u(t)$  within a desired range. There are therefore no differences between goals and constraints in the fuzzy decision framework [28].

**Figure 19.4.** Membership functions representing: (a) fuzzy goal; (b) fuzzy constraints

For each instant of time we have  $n$  fuzzy goals  $G_1, G_2, \dots, G_n$  and  $m$  fuzzy constraints  $C_1, C_2, \dots, C_m$ , and a global objective can be defined as the intersection of every fuzzy membership function. The optimization problem for the entire future control sequence thus becomes

$$\underset{\Delta u}{\text{Max}} \mu_J = \underset{\Delta u}{\text{Max}} (G_1 \cap \dots \cap G_n) \cap (C_1 \cap \dots \cap C_m) \quad (19.5)$$

The maximization of  $\mu_J$  defines a fuzzy decision problem which consists of achieving the highest degree of simultaneous satisfaction of the fuzzy criteria, fuzzy goals and fuzzy constraints. Different operators such as the algebraic product, maximum, sum, or other t-norm can be used to make the fuzzy aggregation. The election of a method must take into account the desired softness in the control outputs, the shape of the membership functions and available computational resources. In [41] it is suggested that the Yager operator be utilized. Because of non-linearity in the process model and/or the decision function, the optimization problem is usually non-convex and cannot be solved using standard algorithms. In order to find the solution of Equation (19.5) using a gradient method such as sequential quadratic programming (SQP), we must relax the Yager operator so that its value is not limited to a fuzzy number between 0 and 1.

The solar power plant in our study is located in Almería, Spain [42] and uses an Acurex distributed collector field. The plant's main objective is to collect solar energy by heating oil passing through the field (see Figure 19.5). The latter consists of 480 collectors arranged in 20 rows that form 10 parallel loops and lie along an east-west axis. It has a tracking system in which the mirrors revolve around an axis parallel to the pipe as they follow the varying inclination of the sun. The cold inlet oil is extracted from the bottom of a storage tank and sent through the field by a pump located at the field inlet. Once heated it is then reintroduced into the storage tank to be used for electricity generation.



**Figure 19.5.** A panoramic view of the Acurex solar collectors

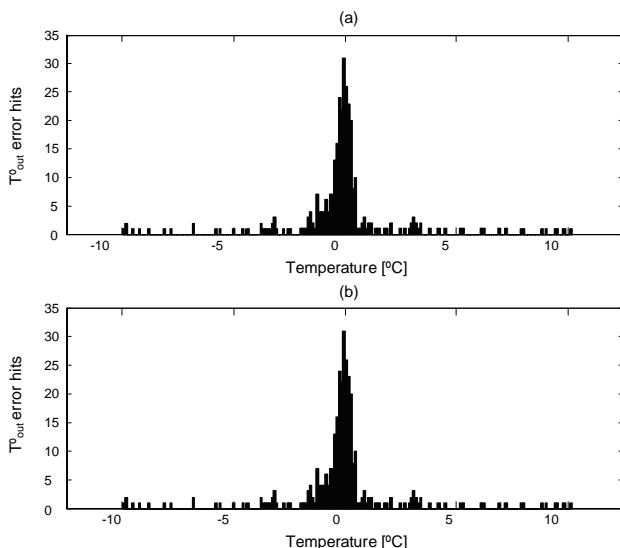
The plant can be described by a set of non-linear distributed parameter equations obtained from the principles of energy and mass balance. A hundred distributed parameter models representing different sections of the collector field are merged to develop a plant simulator.

The main control goal is to maintain constant outlet oil temperature in the face of changes in operating conditions by manipulating the field oil flow. During the

day, when solar conditions are changing, this will require significant flow variations that in turn produce considerable variations in the dynamics of the process. This renders conventional control algorithms based on a simplified model of the process – an LQG regulator, for example – ineffective. Due to its non-linear characteristics, the Acurex solar collector field has been used as an experimental platform for the application of many modern control algorithms [42].

In order to compare fuzzy predictive control with classical predictive control, a simple MPC controller based on a linear model of the solar collector field was designed. The system is kept around the operation point and no hard constraints are imposed on the process. MPC was implemented using a CARIMA linear model for a dynamic system that combines the plant and the feedforward controller [40]. A fuzzy predictive controller was also designed using the same linear model, but with fuzzy characterization of goals for the controlled variable error and constraints on the manipulated variable. The structure of the fuzzy predictive controller is similar to the MPC in the sense that the model and the horizons are the same. Yager t-norm with  $p = 2$  is used as the fuzzy aggregation operator, and the minimization problem is solved using SQP.

The fuzzy characterization of goals and constraints is achieved with fuzzy membership functions  $\mu_e$  for the error (outlet oil temperature minus set-point) and  $\mu_u$  for the variation in the feedforward input temperature  $T_{aux}^o$ . These functions are defined over the set of possible values for both variables, and a general triangular-shaped membership function is defined by three parameters indicating each function's inflection points.



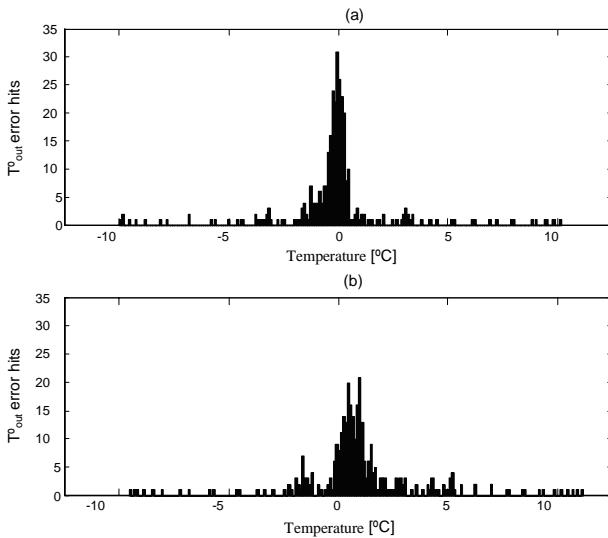
**Figure 19.6.**  $T_{out}^o$  error histograms: (a) Classical MPC; (b) Triangular-shaped fuzzy MPC

The simulation results show that the response of the system with either one of the controllers is the same. The explanation for this result is that the slopes of the fuzzy membership functions were set according to the weights of the classical

MPC. The similarities between the two controllers can be observed in the histograms in Fig. 19.6, which display the  $T^o_{out}$  error distributions for both of them.

As has already been noted, model-based fuzzy predictive control offers several degrees of flexibility, and one of them is the choice of membership functions. A second fuzzy controller that uses custom-shaped membership functions was also studied, assuming in this case that one of the goals of the solar power plant is to maximize the heat transfer from the collector field to an unknown load. On this assumption, the outlet oil temperature set-point is stricter when the temperature is below the set-point, and less strict when the temperature is above it.

The response of the plant using this controller is different from its response with the two controllers considered above. The response times for increments of the set-point are smaller than those obtained when the set-point is decreased. In addition, the manipulated variable is not as oscillatory as with the other two controllers. This behavior is consistent with the fuzzy membership function for the outlet oil temperature error. The modifications on the error profile distribution can be appreciated from the error histogram in Figure 19.7. The reduction in negative error and the increase in positive error in the 0–5°C range are concordant with the shape of the custom membership function  $\mu_e$ .



**Figure 19.7.**  $T^o_{out}$  error histograms for: (a) Classical MPC; (b) Custom-shaped fuzzy MPC

## 19.9 Conclusions

Electrical power plants are a fertile area for the application of a variety of advanced control techniques such as model predictive control and fuzzy control. Though both of them generate benefits individually, these benefits are even greater when the two techniques are combined. Fuzzy predictive control offers two alternatives

for implementation: using non-linear models based on fuzzy expert systems, or using fuzzy logic to characterize the objective function and the constraints. Both alternatives have their specific properties and conditions under which they are particularly appropriate.

This combination of automatic control techniques and computational intelligence also shows great potential for handling other significant problems arising with power plants, which no doubt explains the major ongoing research efforts in this area. Among such problems, we may mention those related to the analysis and control (modeling, state estimation, identification, optimal control and fault detection and isolation) of electric power systems using the hybrid system framework [43].

## Acknowledgement

I wish to thank Fondecyt Chile for their support of projects 1990101, 1020741 and 1050684. I am also grateful to the doctoral and master's program students who contributed to the research reported here.

## References

- [1] D. Flynn, *Thermal Power Plant Simulation and Control*, IEE Power & Energy Series 43, 2003.
- [2] M. Cregan and D. Flynn, "Advances in power plant technology", in D. Flynn, *Thermal Power Plant Simulation and Control*, IEE Power & Energy Series 43, 2003, pp 1-16.
- [3] A.W. Ordys, A.W. Pike, M.A. Johnson, R.M. Katebi and M.J. Grimble, *Modelling and simulation of power generation plants*, Springer, 1994.
- [4] H.G. Kwatny and C. Maffezzoni, "Control of electric power generating plants", in W.S. Levine, *The Control Handbook*, CRC Press, 1996, pp 1453-1483.
- [5] D. Poncia, "Multivariable power plant control", in D. Flynn, *Thermal Power Plant Simulation and Control*, IEE Power & Energy Series 43, 2003, pp 179-201.
- [6] D. Sáez, A. Cipriano and A.W. Ordys, *Optimisation of Industrial Processes at Supervisory Level: Application to Control of Thermal Power Plants*, Springer, 2002.
- [7] E.F. Camacho and C. Bordons, *Model Predictive Control*, Springer, 2005.
- [8] S.J. Qin and T.A. Badgwell, "A survey of industrial model predictive control technology", *Control Engineering Practice*, Vol. 11, 2003, pp 733-764.
- [9] M. Amin, G.Y. Masada and B.F. Womack, "Application of a multivariable self-tuning controller to a power plant boiler", *Proceedings American Control Conference*, 1984.
- [10] B.W. Hogg and N.M. El-Rabaie, "Multivariable generalized predictive control of a boiler system", *IEEE Transactions on Energy Conversion*, Vol. 6, 1991, pp 282-288.
- [11] L. Pérez, F.J. Pérez, J. Cerezo, J. Catediano and J.M. Martín-Sánchez, "Adaptive predictive control in a thermal power station", *Proceedings IEEE Conference on Control Applications*, 1994, pp 747-752.
- [12] G. Prasad, G.W. Irwin, E. Swidenbank and B.W. Hogg, "Plant-wide predictive control for a thermal power plant based on a physical plant model", *IEE Proceedings - Control Theory Applications*, Vol. 147, 2000, pp 523-537.

- [13] K. Krueger, M. Rode and R. Franke, "Optimal control for fast boiler start-up based on a nonlinear model and considering the thermal stress on thick-walled components", *Proceedings IEEE Conference on Control Applications*, Mexico City, 2001.
- [14] H. Peng, T. Ozaki, V. Haggan-Ozaki and Y. Toyoda, "A nonlinear exponential ARX model based multivariable generalized predictive control strategy for thermal power plants", *IEEE Transactions on Control Systems Technology*, Vol. 10, 2002, pp 256-262.
- [15] H. Peng, W. Gui, K. Nakano and H. Shioya, "Robust MPC based on multivariable RBF-ARX model for nonlinear systems", *Proceedings IEEE Conference on Decision and Control*, 2005, Sevilla.
- [16] D. Sáez, A. Ordys and M. Grimble, "Design of a supervisory predictive controller and its application to thermal power plants", *Optimal Control Applications & Methods*, Vol. 26, 2005, pp 169-199.
- [17] R. C. Bansal, "Bibliography on the fuzzy set theory applications in power systems (1994-2001)", *IEEE Transactions on Power Systems*, Vol. 18, 2003, pp 1291-1299.
- [18] H. Matsumoto, Y. Eki, S. Nigawara, M. Tokuhira, Y. Suzuki Y. and A. Kaji, "An operator support expert system based on on-line dynamics simulation and fuzzy reasoning for startup schedule optimization in fossil power plants", *IEEE Transactions on Energy Conversion*, Vol. 8, 1993, pp 674-680.
- [19] A. Ben-Abdenour and K.L. Lee, "An autonomous control system for boiler-turbine units", *IEEE Transactions on Energy Conversion*, Vol. 11, 1996, pp 401-406.
- [20] M. Holmes and A. Ray, "Fuzzy damage-mitigating control of a fossil-fueled power plant", *Proceedings American Control Conference*, 1998, Philadelphia.
- [21] R. Garduno-Ramírez and K.Y. Lee, "Wide-range operation of a power unit via feedforward fuzzy control", *IEEE Transactions on Energy Conversion*, Vol. 15, 2000, pp 421-426.
- [22] P. Kallappa and A. Ray, "Fuzzy wide-range control of fossil power plants for life extension and robust performance", *Automatica*, Vol. 36, 2000, pp 69-82.
- [23] M. Si Fodil, P. Siarry, F. Guely and J.-L. Tyran, "A fuzzy rule base for the improved control of a pressurized water nuclear reactor", *IEEE Transactions on Fuzzy Systems*, Vol. 8, 2000, pp 1-10.
- [24] L. Jianghong, C. Laijiu, Z. Suyin and W. Yiqian, "Multivariable fuzzy PID control system for deareator's and condenser's levels in a thermal power unit", *Cybernetics and Systems*, Vol. 33, 2002, pp 483-506.
- [25] W. Wang, H.-X. Li and J. Zhang, "Intelligence-based hybrid control for power plant boiler", *IEEE Transactions on Control Systems Technology*, Vol. 10, 2002, pp 280-287.
- [26] R. Babuska, *Fuzzy Modeling for Control*, Kluwer Academic Publishers, 1998.
- [27] T. Moelbak and J.H. Mortensen, "Steam temperature control", in D. Flynn, *Thermal Power Plant Simulation and Control*, IEE Power & Energy Series 43, 2003, pp 131-160.
- [28] R. Bellman and L.A. Zadeh, "Decision-making in a fuzzy environment", *Management Science*, Vol. 17, 1970, pp B141-B164.
- [29] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, 1985, pp 116-132.
- [30] K. Marcelle, K. Chiang, P. Bonisone and P. Houpt, "Optimal load cycling of large steam turbines", *Proceedings IEEE International Conference on Fuzzy Systems*, 1994, pp 667-672.
- [31] Z. Huaguang and Z. Bien, "Fuzzy system identification and predictive control of load system in power plant", *Proceedings IEEE International Conference on Fuzzy Systems*, 1998, pp 342-347.

- [32] Z. Huaguang and L. Cai, "Multivariable fuzzy generalized predictive control", *Cybernetics and Systems*, Vol. 33, 2002, pp 69-99.
- [33] E.N. Sánchez, M. Moheno, D.A. Suárez and A. Quintero, "Electric power fossil plant start-up combining predictive control and Takagi-Sugeno fuzzy modeling", *Proceedings IEEE International Symposium on Intelligent Control*, Vancouver, 2002, pp 116-120.
- [34] D. Andone and A. Hossu, "Predictive control based on fuzzy model for steam generator", *Proceedings IEEE International Conference on Fuzzy Systems*, 2004, Vol. 3, pp 1245-1250.
- [35] P. Han, G. Wang and D. Wang, "On the application of predictive functional control in steam temperature systems of thermal power plants", *Proceedings IEEE Conference on Decision and Control*, Hawaii, 2003, pp 6559-6564.
- [36] R. Garduno-Ramírez and K.Y. Lee, "Extending plant load-following capabilities", in D. Flynn, *Thermal Power Plant Simulation and Control*, IEE Power & Energy Series 43, 2003, pp 205-242.
- [37] S. Simani, C. Fantuzzi and R.J. Patton, *Model-based Fault Diagnosis in Dynamic Systems using Identification Techniques*, Springer, 2003.
- [38] R.J. Patton, "Fault-tolerant control: The 1997 situation", *Proceedings IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Hull, 1997, pp 1029-1051.
- [39] D. Sáez and A. Cipriano, "Fuzzy modeling of a combined cycle power plant", *Proceedings IEEE International Conference on Fuzzy Systems*, Seoul, 1999, pp 1186-1190.
- [40] A. Flores, D. Sáez, J. Araya, M. Berenguel and A. Cipriano, "Fuzzy predictive control of a solar power plant", *IEEE Transactions on Fuzzy Systems*, Vol. 13, 2005, pp 58-68.
- [41] U. Kaymak, J. Sousa and M. Stness, "Adaptive decision alternatives in fuzzy predictive control", *Proceedings IEEE International Conference on Fuzzy Systems*, Anchorage, 1998, pp 698-703.
- [42] E. F. Camacho, M. Berenguel and F. R. Rubio, *Advanced Control of Solar Plants*, Springer, 1997.
- [43] G. Ferrari-Trecate, E. Gallesteay, P. Letizia, M. Spedicato, M. Morari and M. Antoine, "Modeling and control of cogeneration power plants: a hybrid system approach", *IEEE Transactions on Control Systems Technology*, Vol. 12, 2004, pp 694-705.

---

## Fuzzy Logic for Robots Calibration – Using Fuzzy Interpolation Technique in Modeless Robot Calibration

Ying Bai and Dali Wang

### 20.1 Introduction

The classical model-based method for robot calibration involves setting up a kinematic model for the robot, measuring positions and orientations of the robot end-effector, identifying its kinematic parameters and compensating its pose errors by modifying its joint angles [1]. Most published research results for robot calibration belong to this category. The advantage of a model-based calibration is that a large workspace can be calibrated accurately and all pose errors within the calibrated workspace can be compensated by joint angles. Its disadvantage lies in the fact that the understanding of kinematic modeling and identification processes needs advanced knowledge of robot kinematics, which may pose a challenge to field engineers.

On the other hand, a modeless robot calibration does not go through any kinematic modeling and identification steps. In a pose measuring process, a robot workspace is divided into a sequence of small squares in a 2D case, or cubes in a 3D case with nominal grid points around each cell assumed known. All position errors on the grid points are measured and recorded by moving the robot through all the grid points. These position errors are stored in memory for future usage. With a modeless method, simple error compensation for a target position can be realized by interpolating errors from its neighboring grid points [2]. Its disadvantage, however, is conflict between calibration accuracy and number of grid points. In spite of this, because of its simplicity and effectiveness, the modeless calibration technique is widely adopted in industrial applications.

There are also several different alternative approaches to the model-based and modeless robot calibration methods discussed at the above. Whitney and Shamma [3, 4] reported a non-parametric accuracy compensation method using polynomial approximating functions; however a nominal inverse kinematic model of the robot to be calibrated is still needed to calculate the joint position vector corresponding to the desired position of the robot end-effector. Another non-parametric compensation approach is to divide the workspace of the robot into a sequence of discrete areas or cubic cells and then use a numerical procedure to determine the inverse kinematic solution for each area or cubic cell. The problem of implementing that approach is that a huge amount of memory is needed to store

those inverse kinematic solutions. To solve this problem, James Albus [5, 6] developed a so-called Cerebellar Model Articulation Controller (CMAC) to reduce the memory size. By using this non-parametric compensation method, the workspace of the compensated robot is about a quarter of the normal robot workspace and the accuracy is around 0.12 mm [3].

Most conventional modeless calibration methods use some interpolation techniques, such as linear interpolation, bilinear interpolation and cubic spline interpolation, which are widely utilized in industrial settings. Because actual error distribution of a robot may not satisfy linear or cubic models, the estimated errors would not be as accurate as desired. To improve numerical interpolation accuracy, different approaches have been attempted by researchers. Zhuang and Wu [7] reported a histogram method for estimating the optimal membership function distribution in order to improve the accuracy of measured positions. Carlson and Looney reported a new image interpolation method that refined the bilinear and fuzzy interpolation techniques [8]. Song, Smith and Rizk provided an optimized fuzzy logic controller for the generation of optimal trajectories based on Optimal Control Table (OCT) [9]. Bai and Wang developed a fuzzy interpolation technique to improve the accuracy of robots calibration in the 2D space [10, 11].

In this chapter, a modeless on-line fuzzy interpolation method is developed and implemented to improve the calibration accuracy of the robot in its 3D workspace. A comparison between the compensation results of using the fuzzy interpolation and linear interpolation techniques is made through simulation studies. Three error models, sinusoidal waveform, normal and uniform distributed errors, are tested, simulating practical settings. Because the actual neighboring errors on each grid point are random distributed, and the error surfaces on each cell are also random functions at a certain moment, the traditional membership functions are not suitable for our study (each cell is a cubic that is surrounded by eight grid points). Moreover, the lookup table may not be calculated in advance until the output membership functions that are associated with the actual position errors on grid points are determined during the error measurement process. This means that one cannot use the traditional off-line fuzzy technology to obtain the lookup table for fuzzy error interpolation. The crisp output must be estimated on-line based on the errors at the neighboring grid points, which means that the output membership functions must be dynamic functions of the neighboring errors.

The remainder of the chapter is organized as follows. The fuzzy interpolation method tailored to robot calibration is discussed in 20.2. Section 20.3 provides simulation results to illustrate the merits of using the fuzzy interpolation in modeless robot calibration over the traditional interpolation method. A comparison of model and modeless method is shown in 20.4. The conclusion is given in 20.5.

## 20.2 On-line 3D Fuzzy Interpolation

As has been mentioned, a modeless compensation method involves only two steps: robot pose measuring and error compensation. Compared with the model-based compensation, the kinematic modeling and identification steps are no longer needed for the modeless compensation method. This can significantly save

workload in a robot calibration. It starts with a measurement process, in which position errors of the robot end-effector are measured at all specified grid points on a sequence of pre-determined cubic cells within the robot workspace. A suitable interpolation technique is then applied to fit the target position error based on the position errors of neighboring grid points around the target position. Among several interpolation methods, the linear interpolation technique is a simple and popular one, and is widely adopted by modeless robot calibration.

When using the modeless method to calibrate robots, it is necessary to have some proper measurement device with sufficient accuracy, such as a Coordinate Measurement Machine (CMM), a set of cameras (a stereo camera system) or a laser tracking system, which is used to gather robot position errors as the robot moves its end-effector to all grid points on the pre-determined cubic cell. Figure 20.1 illustrates a laser tracking system that is used to measure robot position errors.

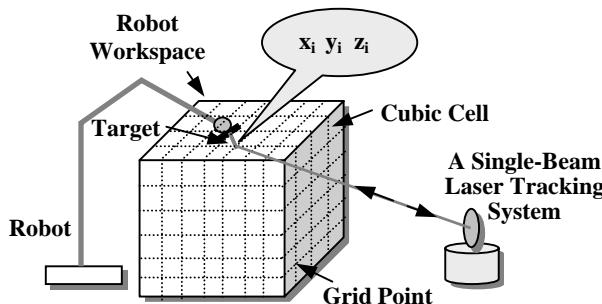
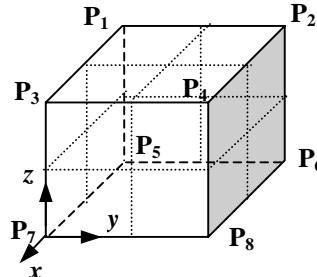


Figure 20.1. A setup for modeless robot calibration

The robot workspace is divided into a sequence of small cubic cells and each small cell is surrounded by eight neighboring grid points, which is shown in Figure 20.2. At each grid point, the laser tracker is used to check the position errors of the end-effector of the robot. In Figure 20.1, the desired position of the grid point  $i$  is  $(x_i, y_i, z_i)$ , and the actual position of the robot end-effector is  $(x_i, y_i, z_i)$ . The position errors for this grid point are  $e_x = x_i - x_i$ ,  $e_y = y_i - y_i$  and  $e_z = z_i - z_i$ . The robot will be moved to all grid points on the robot workspace, and all position errors on these grid points will be measured and stored in the memory for future usage. During the compensation process, as the robot is moved to a certain target point on the workspace, an interpolation technique, say the linear interpolation method, is used to estimate the target position error based on the errors of the neighboring grid points around the target point, and finally these errors are added into the target position to obtain the compensated position, with which, the robot is commanded to move to that position.

Basically, the linear interpolation method is based on the assumption that errors in a cubic cell are distributed uniformly, and the interpolated error is obtained from three planes that are constructed based on eight neighboring errors on the grid points around the target cubic cell. However, in the real world, this assumption may not be valid. Position errors on each cell may not be distributed uniformly. We can consider the  $e_x(x, y, z)$  as the fourth dimension function based on the position  $x$ ,

$y'$  and  $z'$  inside each cubic cell. Same consideration is held for  $e_y(x', y', z')$  and  $e_z(x', y', z')$ . The compensation accuracy of using either the linear or bilinear interpolation technique is limited by this assumption.



**Figure 20.2.** A cubic cell in robot workspace

In order to solve this problem and to improve the compensation accuracy, an on-line fuzzy interpolation method is introduced. The traditional fuzzy inference system uses pre-defined membership functions and control rules to construct lookup tables, and then pick up the associated control output from the lookup table as the fuzzy inference system works in an application. This type of system is called off-line fuzzy inference system because all inputs and outputs have been defined prior to the application process. In this study, this off-line fuzzy system cannot meet the task requirement based on the following reasons: first, the position error of the target point is estimated based on eight errors of the neighboring grid points, and these eight neighboring errors are randomly distributed. The off-line fuzzy output membership functions are defined based on the errors range, in here, the neighboring errors' range. However this range estimation is not as good as the actual errors obtained on the grid points. Second, since one cell needs one lookup table for the off-line fuzzy system, it requires a huge memory to save a large number of lookup tables, which is both space and time consuming and therefore is not suitable for industrial applications. For example, if the robot workspace is divided into  $40 \times 40 \times 40$  small cubic cells, one needs to have 64,000 lookup tables! By using an on-line dynamic fuzzy inference system, one can estimate the target position error on-line by combining the output membership functions that are obtained from the actual errors on the neighboring grid points. With this approach, one doesn't need off-line lookup tables at all. This means that the system determines the output membership functions only after the fuzzy inference system is applied to an actual process, using real errors on the grid points, not a range.

Figure 20.3 shows conceptually the definition of the fuzzy error interpolation inference system. Each small cubic cell that is surrounded by eight neighboring grid points is defined as a cell, and furthermore this cell is further divided equally into eight smaller cells, which are shown in Figure 20.3a. The position error at each grid point is defined as  $P_1, P_2, P_3, P_4, P_5, P_6, P_7$  and  $P_8$ .

For the fuzzy inference system, we apply the interpolation method in three dimensions separately, so the inputs to the fuzzy inference system are  $e_x, e_y$  and  $e_z$ . The outputs are  $ee_x, ee_y$  and  $ee_z$ , which are shown in Figure 20.3b.

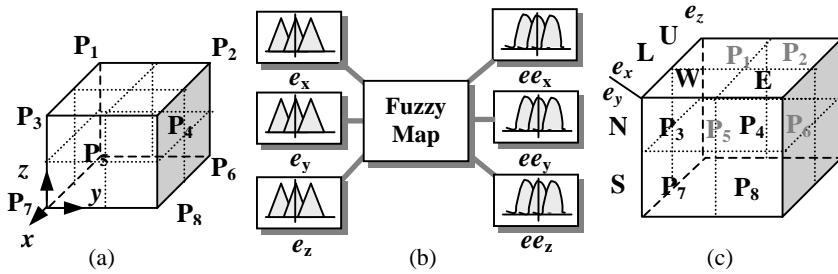


Figure 20.3. Definition of the fuzzy interpolation inference system

The control rule is shown in Figure 20.3c, which is straightforward and based on the human knowledge. The only point to be emphasized is that each  $P_i$  should be considered as a combination of three error components on each grid point,  $P_{xi}$ ,  $P_{yi}$  and  $P_{zi}$ , which are responding to errors in  $x$ ,  $y$  and  $z$  directions.

The distances between the neighboring grid points of each cell on the workspace are 20 mm in  $x$ ,  $y$  and  $z$  directions for this study, which is a standard interval for a mid-size calibration workspace. Totally the workspace includes  $40 \times 40 \times 40$  cells, which is equivalent to an  $800 \times 800 \times 800$  mm<sup>3</sup> space. The input membership functions for  $x$ ,  $y$  and  $z$  directions and the predefined output membership functions are shown in Figure 20.4.

The predefined output membership function is used as a default one, and the final output membership function will be obtained by shifting the default one by the actual error values on the grid points. For each cell, eight output membership functions are implemented and each one is associated with the error at one grid point. In Figure 20.4b, only four output membership functions are shown here because of the space limitation. In an application, total eight membership functions should be utilized.

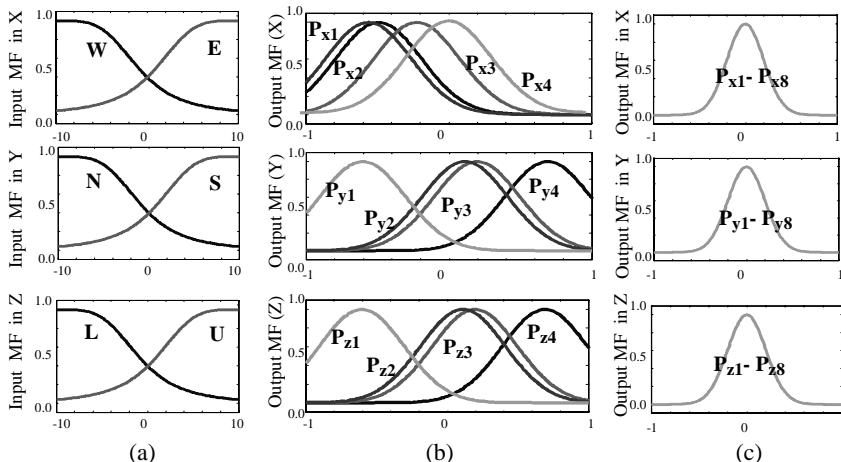


Figure 20.4. Input and output membership functions

The Gaussian-Bell waveforms are selected as the shape of the membership functions for three inputs as shown in Figure 20.4a; the ranges of inputs are between  $-10$  and  $10\text{ mm}$  ( $20\text{ mm}$  interval on grid points). W and E represent the inputs located at different areas in the  $x$  direction, N and S represent the inputs located at different areas in the  $y$  direction, and L and U represent the inputs located at different areas in the  $z$  direction. Unlike the traditional fuzzy inference system, in which all membership functions should be determined to produce the lookup table prior to the implementation of the fuzzy system, in this study the output membership functions will not be defined until the implementation of the fuzzy error mapping to compensate the position errors. So the output membership functions will be determined during the application of the fuzzy inference system on-line or dynamically. This is called dynamic fuzzy system. Figure 20.4b shows an example of the output membership functions, which are related to the simulated random errors at neighboring grid points. Each  $P_{xi}$ ,  $P_{yi}$  and  $P_{zi}$  responds to the position error at the  $i$ th grid point in the  $x$ ,  $y$  and  $z$  directions, respectively. During the design stage, all output membership functions should be initialized to a gaussian waveform with a mean of 0 and a range between  $-0.5$  and  $0.5\text{ mm}$ , which is a typical error range for this workspace in robotic calibration. These output membership functions will be determined on-line based on the errors of the neighboring grid points around the target point in the workspace as mentioned above.

The control rules shown in Figure 20.3c can be interpreted as follows after the output membership functions are determined:

- IF  $e_x$  is W and  $e_y$  is N, and  $e_z$  is U, THEN  $ee_x$  is  $P_{x1}$ ,  $ee_y$  is  $P_{y1}$  and  $ee_z$  is  $P_{z1}$ . ( $P_1$ )
  - IF  $e_x$  is W and  $e_y$  is N, and  $e_z$  is L, THEN  $ee_x$  is  $P_{x3}$ ,  $ee_y$  is  $P_{y3}$  and  $ee_z$  is  $P_{z3}$ . ( $P_3$ )
  - IF  $e_x$  is W and  $e_y$  is S, and  $e_z$  is U, THEN  $ee_x$  is  $P_{x5}$ ,  $ee_y$  is  $P_{y5}$  and  $ee_z$  is  $P_{z5}$ . ( $P_5$ )
  - IF  $e_x$  is W and  $e_y$  is S, and  $e_z$  is L, THEN  $ee_x$  is  $P_{x7}$ ,  $ee_y$  is  $P_{y7}$  and  $ee_z$  is  $P_{z7}$ . ( $P_7$ )
  - IF  $e_x$  is E and  $e_y$  is N, and  $e_z$  is U, THEN  $ee_x$  is  $P_{x2}$ ,  $ee_y$  is  $P_{y2}$  and  $ee_z$  is  $P_{z2}$ . ( $P_2$ )
  - IF  $e_x$  is E and  $e_y$  is N, and  $e_z$  is L, THEN  $ee_x$  is  $P_{x4}$ ,  $ee_y$  is  $P_{y4}$  and  $ee_z$  is  $P_{z4}$ . ( $P_4$ )
  - IF  $e_x$  is E and  $e_y$  is S, and  $e_z$  is U, THEN  $ee_x$  is  $P_{x6}$ ,  $ee_y$  is  $P_{y6}$  and  $ee_z$  is  $P_{z6}$ . ( $P_6$ )
  - IF  $e_x$  is E and  $e_y$  is S, and  $e_z$  is L, THEN  $ee_x$  is  $P_{x8}$ ,  $ee_y$  is  $P_{y8}$  and  $ee_z$  is  $P_{z8}$ . ( $P_8$ )
- (20.1)

The control rules are straightforward and they are based on human knowledge. The error on  $P_1$  grid point should be given more weight if the target position (input) is located inside the NWU area on a cell. A similar conclusion can be derived for errors on all other grid points.

Another issue is the output of the fuzzy inference system. Unlike the off-line fuzzy inference system, the fuzzy inference system implemented in this study is an

on-line one, which means that output of the fuzzy system is not obtained from the pre-defined lookup table that is based on the pre-defined input and output membership functions, but from a real time fuzzy inference calculation that utilizes the pre-defined input membership functions and the real time position errors on the neighboring grid points. The input error variables can be expressed as a label set  $L$ , with  $E$  being a linguistic input variable:

$$L(E) = \{NWU, NWL, NEU, NEL, SWU, SWL, SEU, SEL\} \quad (20.2)$$

Assume that  $u_i$  is the membership function,  $U_i$  the universe of discourse and  $m$  the number of contributions, the traditional output of the fuzzy inference system can be represented as

$$u = \frac{\sum_{i=1}^m (u_i \times U_i)}{\sum_{i=1}^m u_i} \quad (20.3)$$

where  $u$  is the current crisp output of the fuzzy inference system and (20.3) is obtained by using the Center-Of-Gravity method (COG). In this study, both  $u_i$  and  $U_i$  in the output membership functions are random distributed variables and the actual values of these variables depend upon the position errors of eight neighboring grid points around the target position. These relationships can be expressed as:

$$u_{xi} = F_{xi} (P_{x1}, P_{x2}, P_{x3}, P_{x4}, P_{x5}, P_{x6}, P_{x7}, P_{x8}) \quad (20.4)$$

$$U_{xi} = Q_{xi} (P_{x1}, P_{x2}, P_{x3}, P_{x4}, P_{x5}, P_{x6}, P_{x7}, P_{x8}) \quad (20.5)$$

where  $F_{xi}$  is the membership function of the input position in the  $x$  direction and it is a predetermined membership function as shown in Figure 20.4a.  $Q_{xi}$  is the real error output membership function, which is a random distributed function, and it gives the error output contributions in the  $x$  direction. This membership function is determined by the real position errors at the eight neighboring grid points in the  $x$  direction,  $P_{x1} \sim P_{x8}$ . This membership function determines the degree to which the current position input belongs to each different real error output based on the eight control rules defined in (20.1) in the  $x$  direction, and it is equivalent to the universe of discourse or a weighing factor. Placing (20.4) and (20.5) into (20.3), one obtains

$$u_x = \frac{\sum_{i=1}^m F_{xi} (P_{x1}, P_{x2}, \dots, P_{x8}) \times Q_{xi} (P_{x1}, P_{x2}, \dots, P_{x8})}{\sum_{i=1}^m F_{xi} (P_{x1}, P_{x2}, P_{x3}, P_{x4}, P_{x5}, P_{x6}, P_{x7}, P_{x8})} \quad (20.6)$$

Here  $u_x$  represents the final error output of the fuzzy interpolation method in the  $x$  direction. In (20.6),  $Q_{xi}$  will not be determined until the fuzzy error interpolation technique is applied in an actual compensation process, which means that this fuzzy inference system is an on-line process. The final crisp output of the fuzzy error interpolation system is determined by the neighboring position errors of eight grid points. Similar calculations can be implemented to the error outputs in the  $y$  and  $z$  directions.

An algorithm as described above is developed in this study to fulfill this fuzzy error interpolation calculation on-line based on the actual errors on grid points. The actual software includes a high-level control program and an interface to the fuzzy inference system developed in MATLAB [12].

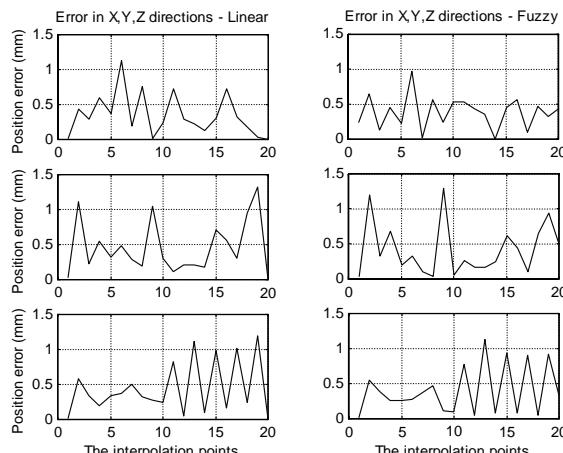
The advantage of using the on-line fuzzy inference system is that the control output has the real time control ability, but the drawback is that this type of control needs a relative longer response time because of the complicated calculation in the fuzzy inference system. This shortcoming becomes little importance with the availability of high speed CPUs for real-time controllers.

### 20.3 MATLAB Simulation Results

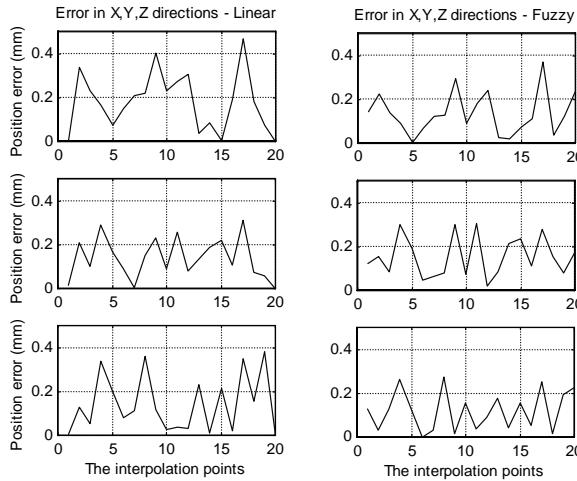
Three different types of error are simulated in this study in order to illustrate the effectiveness of the dynamic on-line fuzzy interpolation technique in comparison to the linear interpolation method. Three types of error are:

- Normal distributed random error
- Uniform distributed random error
- Sinusoidal waveform error

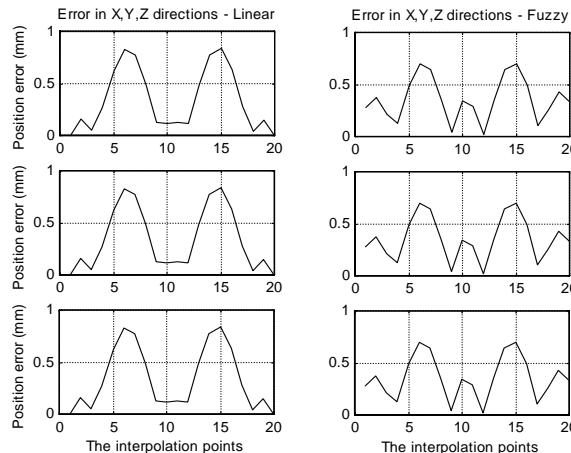
Figures 20.5~20.7 show the simulation results of using both fuzzy interpolation and linear interpolation methods for these three types of error.



**Figure 20.5.** Interpolation results – normal distributed errors



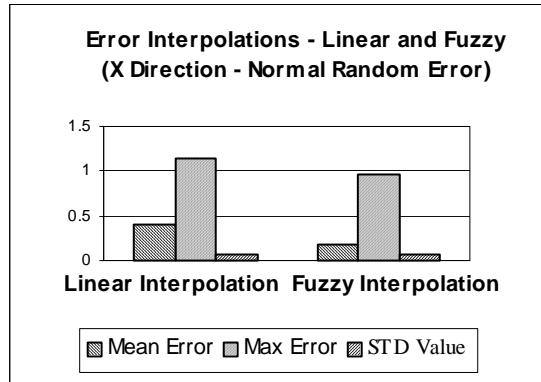
**Figure 20.6.** Interpolation results – uniform distributed errors



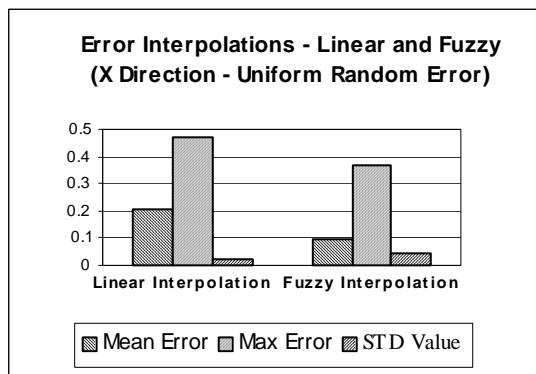
**Figure 20.7.** Interpolation results – sinusoid waveform errors

In Figures 20.5~20.7, three graphics show the differences between the simulated real position errors and the position errors calculated using the linear interpolation technique in the  $x$ ,  $y$  and  $z$  directions, respectively. The simulated target (testing) positions on the robot workspace are scanned from 1 to 20 mm in three directions within one cell with a step of 1 mm. It can be found that all peak errors of the linear interpolation methods are greater than those errors of the fuzzy interpolation technique in all three different error models.

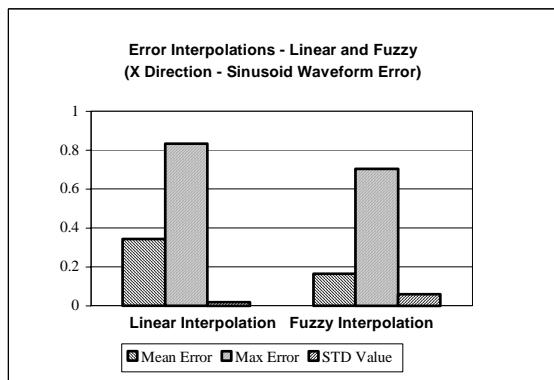
Figures 20.8~20.10 shows the comparison in mean error, max error and STD values represented by using the histograms between the linear interpolation and fuzzy interpolation techniques.



**Figure 20.8.** Normal distributed errors



**Figure 20.9.** Uniform distributed errors



**Figure 20.10.** Sinusoid waveform errors

In Figure 20.8, the comparison of two interpolation techniques used in normal distributed errors is presented. To save the space, only errors in the  $x$  direction is shown here. The same style is used in Figure 20.9 for the comparison of using uniform distributed errors. Figure 20.10 shows the comparison of two interpolations in the presence of sinusoidal waveform errors.

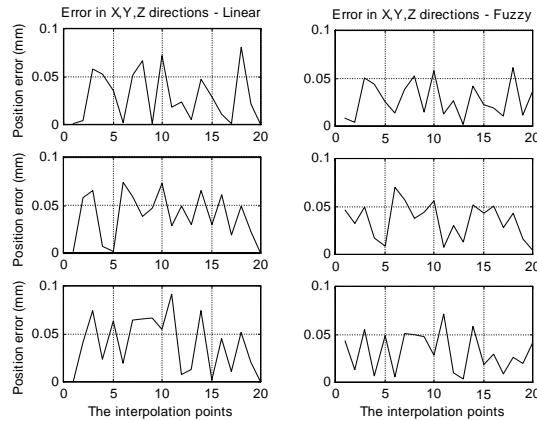
From Figures 20.8~20.10, it can be seen that both mean and maximum errors of the fuzzy interpolation technique are quite smaller than those of linear interpolation method. Approximately the mean errors of the fuzzy error interpolation method are about 40~50% smaller compared with those obtained using the linear interpolation method, and the maximum errors of fuzzy interpolation technique are about 20~40% smaller than those obtained using the linear interpolation technique for all three types of error.

To implement this fuzzy interpolation technique in a real-time application, an interface between the MATLAB and C/C++ or other high level programming languages is needed since the most popular real time control programming language is C/C++, which is used to control the robot to move to all grid points in the robot workspace as well as to measure and record position errors on each grid point, but the fuzzy interpolation method in this study has been developed and implemented in the MATLAB domain [13]. The measured position errors on the grid points can be passed from C/C++ to MATLAB functions, exactly to the fuzzy error interpolation functions using this interface, and in turns the fuzzy error interpolation results can be returned to C/C++ to allow the real time controller to continue to operate for the next target position.

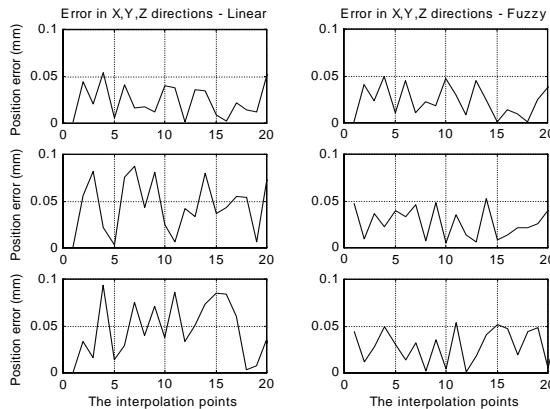
## 20.4 The Comparison Between the Modeless and Model-based Robots Calibration Methods

To compare the performance of modeless and model-based calibration methods, a simulation study is provided and a PUMA 560 robot is utilized to illustrate this implementation. The error model adopted here is MCPC and the simulated position error is  $U[-0.1, 0.1] \text{ mm}$ . The simulated calibration accuracy of model-based method for PUMA 560 robot is about  $0.001 \text{ mm}$ . Depending on the size of the workspace, in this study, the calibration accuracy of using the modeless method can be varied significantly, from  $0.01$  to  $0.0002 \text{ mm}$ . Figures 20.11~20.13 show the comparison between the calibration accuracy of using the modeless and the linear method with the uniform distributed position error in different size of cubic cell in the workspace. The size of the each cubic cell is  $20 \times 20 \times 20 \text{ mm}^3$  in Figure 20.11,  $10 \times 10 \times 10 \text{ mm}^3$  in Figure 20.12, and  $5 \times 5 \times 5 \text{ mm}^3$  in Figure 20.13.

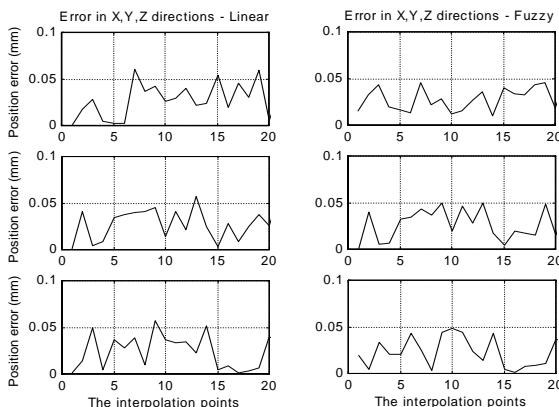
It can be observed that the calibration accuracy of using the fuzzy interpolation technique is higher compared with that of using linear interpolation method. Also it can be seen that the calibration accuracy of using the two methods' identity becomes closer as the cubic cell becomes smaller and smaller as shown in Figure 20.13 when the size of each cubic cell is  $5 \times 5 \times 5 \text{ mm}^3$ . The numbers in the horizontal axes in Figures 20.12 and 20.13 are the number of cells, and the number in the horizontal axis in Figure 20.11 is the simulation step interval, which is from 0 to 20  $\text{mm}$ .



**Figure 20.11.** The comparison of the calibration accuracy ( $20 \times 20 \times 20 \text{ mm}^3$ )

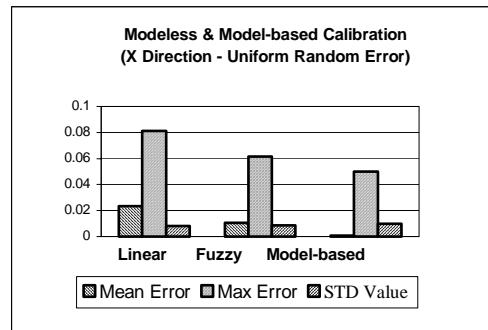


**Figure 20.12.** The comparison of the calibration accuracy ( $10 \times 10 \times 10 \text{ mm}^3$ )

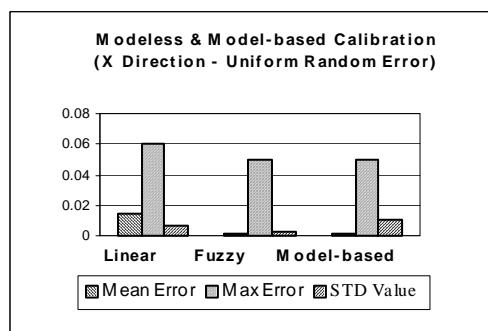


**Figure 20.13.** The comparison of the calibration accuracy ( $5 \times 5 \times 5 \text{ mm}^3$ )

Figure 20.14 shows a comparison between the modeless and model-based calibrations based on the simulated results in histogram. Table 20.1 lists this comparison using the calibration parameters in the data format.

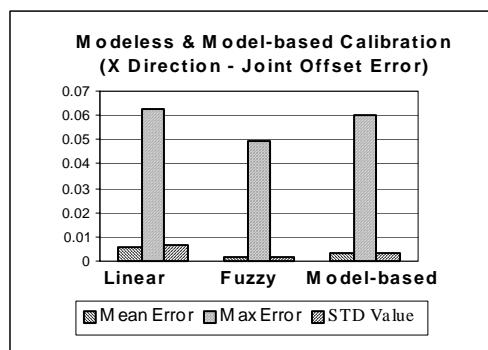


(a) Cubic cell is  $20 \times 20 \times 20 \text{ mm}^3$



(b) Cubic cell is  $10 \times 10 \times 10 \text{ mm}^3$

**Figure 20.14.** A comparison between the modeless and model-based calibration

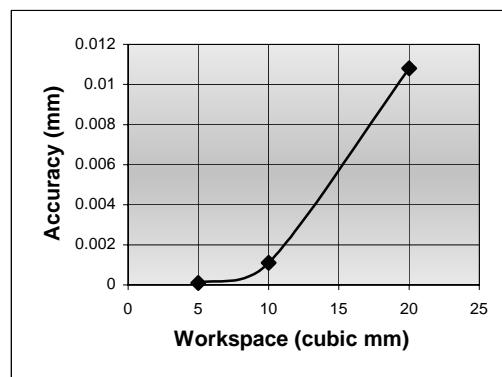


**Figure 20.15.** A comparison of systematic error

**Table 20.1.** Modeless and model-based methods comparison

Cubic cell size	Error Type	Modeless (mm)	Model-based (mm)
$20 \times 20 \times 20 \text{ mm}^3$	Mean	0.0108	0.001
	Max	0.0615	0.05
	STD	0.0087	0.01
$10 \times 10 \times 10 \text{ mm}^3$	Mean	0.0011	0.001
	Max	0.0495	0.05
	STD	0.0020	0.01
$5 \times 5 \times 5 \text{ mm}^3$	Mean	0.0001	0.001
	Max	0.0475	0.05
	STD	0.0002	0.01

Figure 20.16 shows a relationship between the calibration accuracy and the size of the workspace used in the modeless calibration method. It can be found that the modeless calibration accuracy is improved as the workspace becomes smaller and smaller. The values in the  $x$ -axis represent the one-dimensional size of the workspace. For example, a value 5 in the  $x$ -axis means a workspace of  $5 \times 5 \times 5 \text{ mm}^3$ , which is equal to a  $125 \text{ mm}^3$ . Note that in practice, the size of cell depends, among other things, the repeatability of the robot and the accuracy of measurement device. For instance in the case of a PUMA robot, its repeatability is  $0.1 \text{ mm}$ . It is therefore sufficient to choose  $20 \times 20 \times 20 \text{ mm}^3$  as the cell size for a modeless robot calibration as its accuracy is sufficient based on the simulation study (Table 20.1 and Figure 20.16).

**Figure 20.16.** Relationship between the accuracy and workspace

## 20.5 Conclusions

By using the dynamic on-line fuzzy interpolation technique presented in this chapter, the position errors in a modeless robot calibration can be greatly reduced. The effectiveness of using the proposed technique is confirmed by the simulation studies, in which three typical error models, sinusoid waveform, normal distributed and uniform distributed errors, are utilized. This fuzzy interpolation technique is suitable for the modeless robot position compensation. One drawback of the proposed method is its computational complexity in comparison to the linear interpolation method. With the advance of real-time operating systems, the difficulty due to this problem for on-line applications will be greatly reduced.

## References

- [1] B. W. Mooring, Z. S. Roth and M. R. Driels, *Fundamentals of Manipulator Calibration*, Wiley, 1991.
- [2] Hanqi Zhuang and Z. S. Roth, *Camera-Aided Robot Calibration*, CRC Press, Inc. 1996.
- [3] J. S. Shamma and D. E. Whitney, A method for inverse robot calibration. *Journal of Dynamic Systems, Measurement, and Control*, 109 (1): 36-43, March 1987.
- [4] J. S. Shamma, *A Method for Inverse Robot Calibration*, Master's thesis, Massachusetts Institute of Technology, January 1985.
- [5] J. S. Albus, "Data storage in the cerebella model articulation controller (CMAC)," *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, 97:228-233, September 1975.
- [6] J. S. Albus, "A new approach to manipulator control: The cerebella model articulation controller (CMAC)," *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, 97:220-227, September 1975.
- [7] H. Zhuang and Xiaomin Wu, "Membership function modification of fuzzy logic controllers with histogram equalization," *IEEE Trans. On System, Man, and Cybernetics – Part B: Cybernetics*, Vol. 31, No. 1, February 2001, pp 125 – 131.
- [8] B. Carlson and C. Looney, "A new method of image interpolation," *Proceedings of the ISCA 16th International Conference on Computers and their Applications*, Seattle Washington, March 28-30, 2001, pp 25-28.
- [9] F. Song, S. M. Smith, and C. G. Rizk, "A fuzzy logic controller design methodology for 4D system with optimal global performance using enhanced cell state space based best estimate directed search method," *IEEE Int. Conf. On Systems, Man and Cybernetics*, Tokyo, Japan, Oct 12, 1999, pp 138-143.
- [10] Ying Bai and Dali Wang, "Improve the position measurement accuracy using a fuzzy error interpolation technique," *Proceedings Of The IEEE 2003 International Symposium On Computational Intelligence For Measurement Systems and Applications*. July 29-31, 2003, Lugano, Switzerland, pp 227 – 232.
- [11] Ying Bai and Dali Wang, "On the comparison of interpolation techniques for robotic position compensation," *2003 IEEE International Conference on Systems, Man and Cybernetics*, October 5-8, 2003, Washington DC, pp 3384 – 3389.
- [12] Ned Gulley and J.-S. Roger. Jang, *Fuzzy Logic Toolbox User Guide*, The MathWorks, Inc. 1995.
- [13] Ying Bai, *Applications Interface Programming Using Multiple Languages – A Windows Programmer's Guide*, Prentice Hall, March 2003.

## **Fuzzy Control on Manufacturing Welding Systems: To Apply Fuzzy Theory in the Control of Weld Line of Plastic Injection-Molding**

Mei-Yung Chen, Yi-Cheng Chen, and Shia-Chung Chen

### **21.1 Introduction**

In plastic injection molded parts, a weld line is one of the defects occurring; the line affects both the appearance and the strength of the products [1,2]. In the injection molding process, weld lines will inevitably be present as a result of geometrical changes in molded parts. Therefore, in molding design, adjusting designing parameters (such as gate position, products thickness, and so on) can move the weld line in the optimum position, *i.e.*, the symmetrical or least stressed position [3]. However, adjusting the parameters requires rich experience to make good judgments. Even an experienced technician cannot modify the parameters quickly and accurately to place the weld line to the optimum position. Although present CAE package software can provide much information before actual molds were made, the process of simulation trials is time-consuming. In other words, CAE package software rightly needs to establish optimum design and control rules to adjust the parameters in determining the weld-line position in the shortest possible time.

Since 1968, fuzzy theory has been widely accepted ever since Prof. Zadeh published the theory of fuzzy sets [4, 5]. Prof. Zadeh first proposed mathematical models with fuzzy sets to describe vague objects; he also argued that the traditional crisp set theory is not suitable in dealing with complex systems centered on humans and the like. The theory proposed by Prof. Zadeh has inspired Prof. Mamdani to use IF~THEN to establish fuzzy production rules, with which “Fuzzy Theory” was successfully applied in controlling steam engines [6]. From then on, fuzzy control has been investigated in various fields with good results [7, 8].

In the case covered here, fuzzy control will best suit the need. This study tried to transform expert experience into IF~THEN rules to establish fuzzy production rules, which were then applied in controlling the position of the weld line. It is hoped that the time for trial-and-error in using CAE package software can be cut down, thereby facilitating a faster designing process.

## 21.2 Fuzzy Control Logic for the Weld Line Position

Fuzzy control, which is based on the theory of fuzzy logic, expresses expert experience as inference rules and then establishes a controller which can determine the output of the system. The framework of fuzzy control can be divided into four sections (Figure 21.1), which are described separately below:

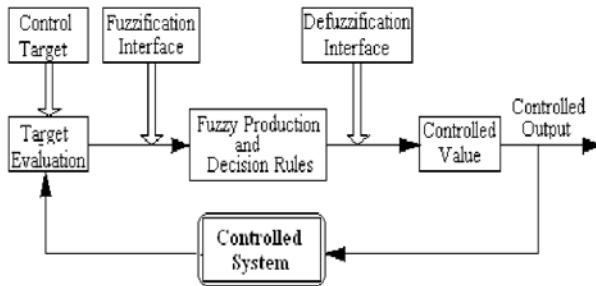


Figure 21.1. Illustration of framework of fuzzy control

### 21.2.1 Fuzzification Interface

The objective of fuzzification interface is to fuzzify input signals as a membership function which establishes a functional relationship between input signals and fuzzy values. The relationship between input signal and fuzzy set can be represented as a curve by input signal (as abscissa) and the grade of membership (as ordinate), as shown in Figure 21.2.

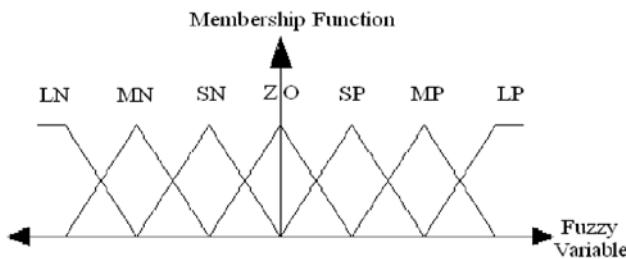


Figure 21.2. Membership function diagram

### 21.2.2 Production Rule

Production rule is the fuzzy logic relation which directly determines the function of the controller. Good production rules lead to a better performance of the controller. The dialogue expressed by fuzzy condition is shown as follows:

***IF E is LN or SN and V is ZO, then U is SP,***

where  $E$  is an input signal,  $V$  is a variation of signal,  $U$  is a controlled output, and  $LN$ ,  $SN$ , and  $SP$  are the fuzzification of input and output values.

According to expert experience, every rule is established and summarized in Table 21.1.

**Table 21.1.** A summarization of control rules

$U$		$V$				
		$LN$	$SN$	$ZO$	$SP$	$LP$
$E$	$LN$	$LP$	$LP$	$LP$	$LP$	$LP$
	$SN$	$LP$	$SP$	$SP$	$ZO$	$ZO$
	$ZO$	$SP$	$SP$	$ZO$	$SN$	$LN$
	$SP$	$ZO$	$ZO$	$SN$	$SN$	$LN$
	$LP$	$LN$	$LN$	$LN$	$LN$	$LN$

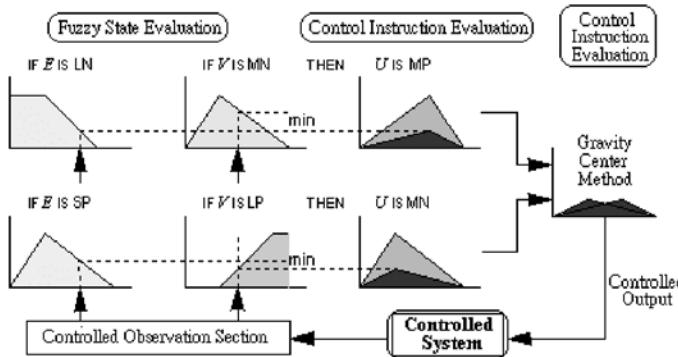
where

- $LP$  = Large Positive
- $MP$  = Medium Positive
- $SP$  = Small Positive
- $ZO$  = Zero
- $SN$  = Small Negative
- $MN$  = Medium Negative
- $LN$  = Large Negative

### 21.2.3 Decision Making Logic

In the step of fuzzy inference, the fuzzy inference operation uses possibly suitable control rules to calculate the output value after actual input is obtained. There are many inference decision methods, such as the center of gravity, area, height, and so on. So far the center of gravity method is considered an acceptable way of inferential decision [4]. The method is illustrated in Figure 21.3.

The evaluation of the fuzzy state in the first part is to calculate the membership functions of the rules with given inputs,  $E$  and  $V$ . The evaluation of the control statements in the second part is to calculate the inferential result  $U$ , in order to find the weightings of the statements. The control statements in the third part summarize the results obtained and their center of gravity was then used to calculate the overall inferential conclusion of the rules.



**Figure 21.3.** The center-of-gravity inference process for state evaluation in fuzzy control

#### 21.2.4 Defuzzification Interface

To perform an ordinary control, a defuzzified control value is needed. In other words, the vague output signal is transformed into a definite value to perform an actual control statement. The method of gravity was employed here and can be expressed as

$$U_{out} = \frac{\int u_o(X_i) \times X_i dX_i}{\int u_o(X_i) dX_i}, \quad (21.1)$$

where  $U_{out}$  is the input of fuzzy control,  $u_o(X_i)$  is the membership grade of output value generated by the  $i$ th rule, and  $X_i$  is the output value generated by the  $i$ th rule. Following the above steps, fuzzy control can be applied in real applications.

A CAE package software, *C-Mold*<sup>®</sup>, was used to find the weld line's position [8], the coordinates of which was regarded as reference for input variables.

### 21.3. Case Analyses

Figure 21.4 shows a molded part; the corresponding nodes pattern constructed by the finite element method is shown in Figure 21.5. The thicknesses of blocks A, B, C, D are equal to 4, 2, 4, and 3 mm, respectively. To move the weld line to the center (80 mm) of the product is the designated goal of control. However, because of gate and other factors, the weld line can only be altered 20 mm from the center to left and right, respectively. The thickness requirement of the right of the product (B block) limits the thickness from 0.5 mm more to 0.2 mm less. First, we adjusted the position of the gate to conform the designated goal of control. If unsuccessful,

the thickness of the right of the product was then adjusted to move the weld line to the demanded position.

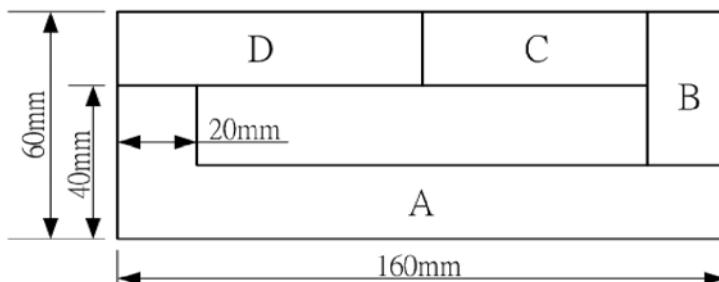
The input and output variables were first normalized to simplify the setting of membership functions. Triangle fuzzy variables were used to express the membership function, and its expression is

$$\mu_m(x) = \frac{1}{a}(-|x-b|+a) \vee 0, \quad a \neq 0. \quad (21.2)$$

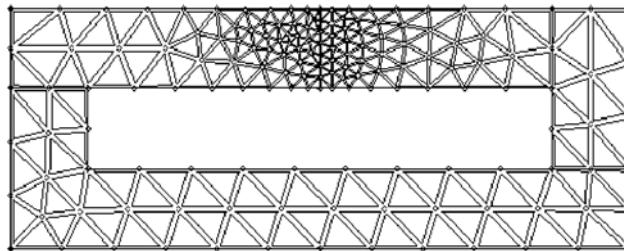
Triangular calculation is easier than and its performance is comparable with other membership functions [4]. The triangular fuzzy set is shown in Figure 21.2.

- 1) The input variable and its membership function density:  $X$  was designated to represent the coordinate variable of the weld line position. The initial position of  $X$  was set to be the large positive number whose membership function is shown in Figure 21.6.
- 2) The output variable and its membership function density
  - a)  $U$  was designated to represent the coordinate variable of the gate, whose left coordinate was set to be the large negative value and whose right coordinate, the large positive value. Their membership functions are shown in Figure 21.6.
  - b)  $V$  was designated to represent the thickness variable of the right of the molded part (Block B), whose upper-limit coordinate was set to be the large negative value and whose right-limit coordinate, the large positive value. Their membership functions are shown in Figure 21.6.

Expert experience and knowledge was transformed into IF~THEN rules. First, the relationship rules for the positions of gate and weld line are established and shown below.



**Figure 21.4.** Initial dimension of the molded part

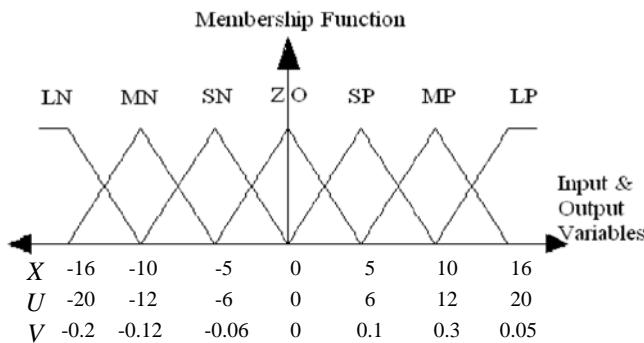


**Figure 21.5.** Final dimension of the molded part

Assumption:  $X$  and  $U$  represent the variables of the position coordinates of the weld line and gate, respectively.

Rule (I):

- 1 IF  $X$  is PB, THEN  $U$  is PB
- 2 IF  $X$  is PM, THEN  $U$  is PM
- 3 IF  $X$  is PS, THEN  $U$  is PS
- 4 IF  $X$  is PB, THEN  $U$  is PB and the design is completed
- 5 IF  $X$  is NS, THEN  $U$  is NS
- 6 IF  $X$  is NM, THEN  $U$  is NM
- 7 IF  $X$  is NB, THEN  $U$  is NB
- 8 IF  $U$  exceeds the left or right boundary, THEN  $U$  is the left or right boundary



**Figure 21.6.** The distributions of  $X$ ,  $U$ , and  $V$  membership functions

It was followed by establishing the rules for the thickness of the right of the molded part (Block B) and the position of the weld line.

Assumption:  $V$  represents the thickness variable of the right of the molded part.

Rule (II):

- 1 IF  $X$  is PB, THEN  $U$  is PB
- 2 IF  $X$  is PM, THEN  $V$  is PM
- 3 IF  $X$  is PS, THEN  $V$  is PS
- 4 IF  $X$  is ZO, THEN  $V$  is ZO and the design is completed

- 5 IF  $X$  is NS, THEN  $V$  is NS
- 6 IF  $X$  is NM, THEN  $V$  is NM
- 7 IF  $X$  is NB, THEN  $V$  is NB
- 8 IF  $V$  exceeds the upper or lower limit, THEN  $V$  is the upper or lower limit

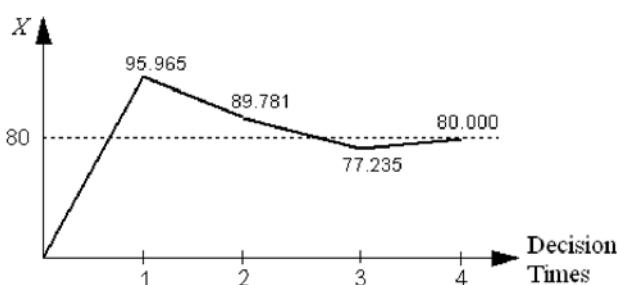
The outputs of rule (I) were used to calculate decision logic with the gravity method, and the decision logic determined a controlled value to move the position of the gate. If the gate position exceeds the left or right limits and the weld line does not reach targeted position, the process enters rule (II). The weld line position obtained in rule (I) was taken as the large (positive or negative) output variable  $X$  of rule (II) to control the weld line position by altering the thickness of the right of the molded part (Block B). The decision method is as Rule (I).

## 21.4 Simulation and Experimental Results

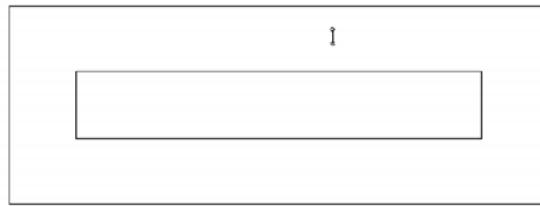
Table 21.2 summarizes the variations of inputs and outputs by applying fuzzy control; the outputs variations are also shown in Figure 21.7. The molded part produced according to the information of the weld line from the above analysis is shown in Figure 21.12.

**Table 21.2.** Variations of inputs and outputs of fuzzy control

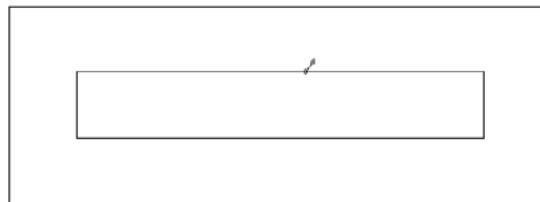
Decision times	Gate position $U$ (mm)	B block thickness $V$ (mm)	Weld line position $X$ (mm)	Result
1	80.0	2.0	95.965	Figure 21.8
2	100.0	2.0	89.781	Figure 21.9
3	100.0	2.5	77.235	Figure 21.10
4	100.0	2.333	80.000	Figure 21.11



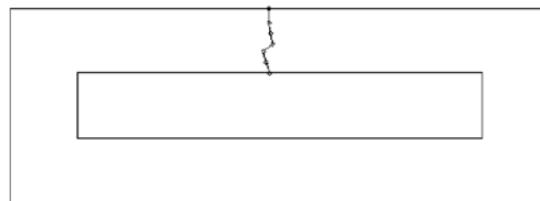
**Figure 21.7.** The outputs variation of fuzzy control



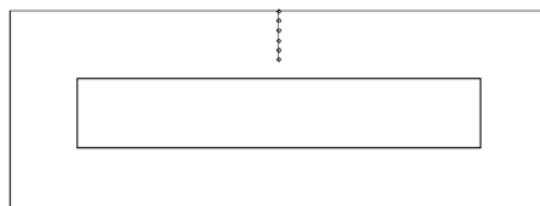
**Figure 21.8.** The originally set position ( $U=80$  mm,  $V=2$  mm) of the weld line



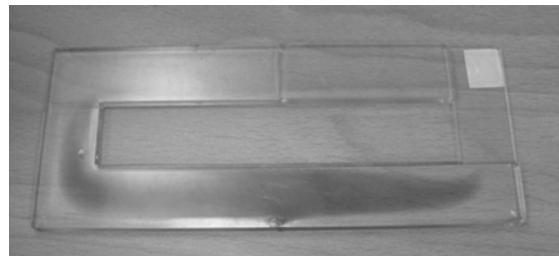
**Figure 21.9.** The second set position ( $U=100$  mm,  $V=2$  mm) of the weld line



**Figure 21.10.** The third set position ( $U=100$  mm,  $V=2.5$  mm) of the weld line



**Figure 21.11.** The fourth set position ( $U=100$  mm,  $V=2.333$  mm) of the weld line



**Figure 21.12.** Photograph of the produced molded part

## 21.5 Conclusion

The present study has successfully applied fuzzy control logic in controlling the weld line position for an injection-molded part. Although only a simple molded part was dealt with here, the incorporation of fuzzy control with CAE software in controlling the weld line position is a breakthrough in the concept of mold-design optimization. During the analysis, only four calculations of decision and simulation were conducted before the final result was obtained, saving considerable time compared with conventional trial and error or computer simulation methods. Consequently, the mold-design process can be accelerated. Since the weld line does appear at the expected location in the actual molded part, the present results can provide a sound basis for developing mold-design optimization.

Future works under consideration include:

- 1) Apply the present approach to molded parts with complex geometry.
- 2) Apply the present approach to molded parts with two or more gates, *i.e.*, two or more output variables are necessary, as

**IF  $E$  is MP and  $V$  is NP, THEN  $U_1$  is SP and  $U_2$  is NP**

- 3) Membership functions and some rules are established by neural networks so as to reduce manual input.

## Acknowledgement

This research is sponsored by National Science Council, R.O.C., under the grant NSC-93-2218-E-157-001.

## References

- [1] W. R. Jong and K. K. Wang, "Automatic Interpretation of Weld Lines and Air Traps in Injection Molding by Synthesizing Deterministic and Heuristic Knowledge," *ANTEC*, May 7-11, 1990, pp 197-202.
- [2] I. S. Dairanieh, A. Haufe, H. J. Wolf and G. Mennig, "Computer Simulation of Weld Lines in Injection Molding Poly (Methyl Methacrylate)," *Polymer Engineering and Science*, Vol. 36, No. 15, 1996, pp 2050-2057.
- [3] V. W. S. Yeung, and K. H. Lau, "Injection Moulding, C-Mold CAE Package, Process Parameter Design and Quality Function Deployment: A Case Study of Intelligent Materials Processing," *Journal of Materials Processing Technology*, Vol. 63, No. 1, 1997, pp 481-487.
- [4] L.A. Zadeh, "Fuzzysets," *Information and Control* , Vol. 8, 1965, pp.338-353.
- [5] L.A. Zadeh, "Outline of a New Approach to the Analysis Complex Systems and Decision Processes," *IEEE Trans. Syst. Man. Cybern.*, Vol. SMC-3, 1973, pp 28-44.
- [6] E. H. Mamdan, "Twenty Years of Fuzzy Control: Experiences Gained and Lessons Learnt," *Second IEEE International Conference on Fuzzy Systems*, Vol. 1, April, 1993, pp 339-344.
- [7] C. C. Lee, "Fuzzy Logic in Control Systems, Fuzzy Logic Controller-Part I," *IEEE Trans. Syst. Man. Cybern.*, Vol. 20, No. 2, Mar./Apr. , 1990, pp 404-418.
- [8] C. C. Lee, "Fuzzy Logic in Control Systems", Fuzzy Logic Controller-Part II," *IEEE Trans. Syst. Man. Cybern.*, Vol. 20, No. 2, Mar./Apr., 1990, pp 419-435.

---

# Index

## A

actuation layer, 132  
adaptive control, 43  
adaptive cruise control, 129  
adaptive FLC, 43  
adaptive neural networks, 237  
adaptive predictive model, 283  
aerial vehicles, 240  
aerodynamic forces, 209  
aerodynamics analysis, 208, 220  
alternating filter, 117  
altitude control, 240  
amplitude distribution, 177  
analytical control surface, 179, 185  
angular velocity, 92, 208  
ARMAX model, 283  
artificial neural networks, 115  
association, 251  
attitude angle, 208, 217  
aumann mean, 118-120  
automated manual transmission, 145, 157  
automatic formation flight control, 240  
automatic thresholding, 117  
automatic transmissions, 145  
autonomous robots, 130  
autonomous underwater vehicle, 191  
AUTOPIA, 130  
autopilot signal, 228

average brightness, 102-103

## B

backpropagation algorithm, 159  
backward driving mode, 163, 166, 168-169  
Baddeley-Molchanov average, 124  
Baddeley-Molchanov fuzzy set average, 120-121, 123  
Baddeley-Molchanov mean, 118, 119-121  
balance position, 98  
bang-bang controller, 198  
BCFPI, 61-63  
bell-shaped waveform, 25  
beta distributions, 122  
Bezier curve, 56, 59, 63-64  
Bezier Curve Fuzzy PI controller, 61  
Bezier function, 54  
bilinear interpolation, 90, 300, 302  
binary classifier, 253  
Bo105 helicopter, 208  
body frame, 238  
boiler following mode, 280, 283  
border pixels, 101  
boundary layer, 192-193, 195-198  
boundary of a fuzzy set, 26  
braking resistance, 265  
bumpy control surface, 55

**C**

CAE package software, 315, 318  
 calibration accuracy, 83, 299-300,  
 309, 310, 312  
 CARIMA models, 290  
 case-based reasoning, 253  
 center of gravity method, 29-30, 32-  
 33  
 centroid defuzzification, 7  
 centroid defuzzification, 56  
 centroid Method, 106  
 characteristic polygon, 57  
 characterization, 43, 251, 293  
 chattering, 6, 84, 191-192, 195, 196,  
 198  
 chromosomes, 59  
 circuit breaker, 270  
 classical control, 1  
 classical set, 19-23, 25-26, 36, 254  
 classification, 106, 108, 111, 179,  
 185, 251-253  
 classification model, 253  
 close formation flight, 237  
 close path tracking, 223-224  
 clustering, 104, 106, 108, 251-253,  
 255, 289  
 clustering algorithm, 252  
 clustering function, 104  
 clutch stroke, 147  
 coarse fuzzy logic controller, 94  
 collective pitch angle, 209  
 collision avoidance, 166, 168  
 collision avoidance system, 160,  
 167, 169-170, 172  
 collision avoidance system, 168  
 complement, 20, 23, 45  
 compressor contamination, 289  
 conditional independence graph, 259  
 confidence thresholds, 251  
 confidence-rated rules, 251  
 coning angle, 210  
 constant gain, 207  
 constant pressure mode, 280  
 contrast intensification, 104  
 contrast intensifier operator, 104  
 control derivatives, 211  
 control gain, 35, 72, 93, 96, 244

control gain factor, 93  
 control gains, 53, 226  
 control rules, 18, 27, 28, 35, 53, 64,  
 65, 90-91, 93, 207, 228, 230, 262,  
 302, 304-305, 315, 317  
 control surfaces, 53-55, 64, 69, 73,  
 77, 193  
 controller actuator faulty, 289  
 control-weighting matrix, 207  
 convex sets, 119-120  
 Coordinate Measurement Machine,  
 301  
 coordinate measuring machine, 96  
 core of a fuzzy set, 26  
 corner cube retroreflector, 85  
 correlation-minimum, 243-244  
 cost function, 74-75, 213, 282-283,  
 287  
 coverage function, 118  
 crisp input, 18, 51, 182  
 crisp output, 7, 34, 41-42, 51, 184,  
 300, 305-306  
 crisp sets, 19, 21, 23  
 crisp variable, 18-19, 29  
 critical clearing time, 270  
 crossover, 59  
 crossover probability, 59-60  
 cruise control, 129-130, 132-135,  
 137-139  
 cubic cell, 299, 301-302, 309  
 cubic spline, 48  
 cubic spline interpolation, 300  
 current time gap, 136  
 custom membership function, 294  
 customer behavior, 249

**D**

damping factor, 211  
 data cleaning, 250  
 data integration, 250  
 data mining, 249, 250, 251-255,  
 259-260  
 data selection, 250  
 data transformation, 250  
 d-dimensional Euclidean space, 117,  
 124  
 decision logic, 321  
 decomposition, 173, 259

defuzzification function, 102, 105, 107-108, 111  
 defuzzifications, 17-18, 29, 34  
 defuzzifier, 181, 242  
 density function, 122  
 dependency analysis, 258  
 dependency structure, 259  
 dependent loop level, 279  
 depth control, 202-203  
 depth controller, 202  
 detection point, 169  
 deviation, 79, 85, 185-188, 224, 251, 253, 262, 265, 268, 276, 288  
 dilation, 117  
 discriminated rules, 251  
 discrimination, 251, 252  
 distance function, 119-121  
 distance sensor, 167, 171  
 distribution function, 259  
 domain knowledge, 254-255  
 domain-specific attributes, 251  
 Doppler frequency shift, 87  
 downhill simplex algorithm, 77, 79  
 downwash, 209  
 drag reduction, 244  
 driver's intention estimator, 148  
 dutch roll, 212  
 dynamic braking, 261-262  
 dynamic fuzzy system, 286, 304  
 dynamic tracking trajectory, 98

**E**

edge composition, 108  
 edge detection, 108  
 eigenvalues, 6-7, 212  
 electrical coupling effect, 85, 88  
 electrical coupling effects, 87  
 equilibrium point, 207, 216  
 equivalent control, 194  
 erosion, 117  
 error rates, 96  
 estimation, 34, 53, 119, 251, 283, 295, 302  
 Euler angles, 208  
 evaluation function, 258  
 evolution, 45, 133, 208, 251  
 execution layer, 262-266, 277  
 expert knowledge, 160, 191, 262

expert segmentation, 121-122  
 extended sup-star composition, 182

**F**

fault accommodation, 284  
 fault clearing states, 271, 274  
 fault detection, 288-289, 295  
 fault diagnosis, 284  
 fault durations, 271, 274  
 fault isolation, 284, 288  
 fault point, 270-271, 273-274  
 fault tolerant control, 288  
 fault trajectories, 271  
 feature extraction, 256  
 fiber glass hull, 193  
 fin forces, 210  
 final segmentation, 117  
 final threshold, 116  
 fine fuzzy controller, 90  
 finer lookup table, 34  
 finite element method, 318  
 finite impulse responses, 288  
 firing weights, 229  
 fitness function, 59-60, 257  
 flap angles, 209  
 flight aerodynamic model, 247  
 flight envelope, 207, 214, 217  
 flight path angle, 210  
 flight trajectory, 208, 223  
 footprint of uncertainty, 176, 179  
 formation geometry, 238, 247  
 formation trajectory, 246  
 forward driving mode, 163, 167, 169  
 forward flight control, 217  
 forward flight speed, 217  
 forward neural network, 288  
 forward velocity, 208, 214, 217, 219-220  
 forward velocity tracking, 208  
 fossil power plants, 284-285, 296  
 four-dimensional synoptic data, 191  
 four-generator test system, 269  
 Fourier filter, 133  
 four-quadrant detector, 79, 87, 92, 96  
 foveal avascular zone, 123  
 fundus images, 115, 121, 124  
 fuselage, 208-210

fuselage axes, 208-209  
 fuselage incidence, 210  
 fuzz-C, 45  
 fuzzifications, 18, 25  
 fuzzifier, 181-182  
 fuzzy ACC controller, 138  
 fuzzy aggregation operator, 293  
 fuzzy ASICs, 37-38, 50  
 fuzzy binarization algorithm, 110  
 fuzzy CC controller, 138  
 fuzzy clustering algorithm, 106, 108  
 fuzzy constraints, 286, 291-292  
 fuzzy control surface, 54  
 fuzzy damage-mitigating control, 284  
 fuzzy decomposition, 108  
 fuzzy domain, 102, 106  
 fuzzy edge detection, 111  
 fuzzy error interpolation, 300, 302, 305-306, 309, 313  
 fuzzy filter, 104  
 fuzzy gain scheduler, 217-218  
 fuzzy gain-scheduler, 207-208, 220  
 fuzzy geometry, 110-111  
 fuzzy I controller, 76  
 fuzzy image processing, 102, 106, 111, 124  
 fuzzy implication rules, 27-28  
 fuzzy inference system, 17, 25, 27, 35-36, 207-208, 302, 304-306  
 fuzzy interpolation, 300, 302, 305-307, 309, 313  
 fuzzy interpolation method, 309  
 fuzzy interpolation technique, 300, 309, 313  
 fuzzy interval control, 177  
 fuzzy mapping rules, 27  
 fuzzy model following control system, 84  
 fuzzy modeling methods, 255  
 fuzzy navigation algorithm, 244  
 fuzzy operators, 104-105, 111  
 fuzzy P controller, 71, 73  
 fuzzy PD controller, 69  
 fuzzy perimeter, 110-111  
 fuzzy PI controllers, 61  
 fuzzy PID controllers, 53, 64-65, 80

fuzzy production rules, 315  
 fuzzy reference governor, 285  
 Fuzzy Robust Controller, 7  
 fuzzy set averages, 116, 124-125  
 fuzzy sets, 7, 19, 22, 24, 27, 36, 45, 115, 120-121, 124-125, 151, 176-182, 184-188, 192, 228, 262, 265-266  
 fuzzy sliding mode controller, 192, 196-197  
 fuzzy sliding surface, 192  
 fuzzy subsets, 152, 200  
 fuzzy variable boundary layer, 192  
 fuzzyTECH, 45

**G**

gain margins, 207  
 gain scheduling, 193, 207, 208, 211, 217, 220  
 gas turbines, 279  
 Gaussian membership function, 7  
 Gaussian waveform, 25  
 Gaussian-Bell waveforms, 304  
 gear position decision, 145, 147  
 gear-operating lever, 147  
 general window function, 105  
 general-purpose microprocessors, 37-38, 44  
 genetic algorithm, 54, 59, 192, 208, 257-258  
 genetic operators, 59-60  
 genetic-inclined search, 257  
 geometric modeling, 56  
 gimbal motor, 90, 96  
 global gain-scheduling, 220  
 global linear ARX model, 284  
 global navigation satellite systems, 141  
 global position system, 224  
 goal seeking behaviour, 186-187  
 governor valves, 280

**H**

Hamiltonian function, 261, 277  
 hard constraints, 283, 293  
 heading angle, 226, 228, 230, 239, 240-244, 246  
 heading angle control, 240

- heading controller, 194, 201-202  
 heading error rate, 194, 201  
 heading speed, 226  
 heading velocity control, 240  
 heat recovery steam generator, 279  
 hedges, 103-104  
 height method, 29  
 helicopter, 207-212, 214, 217, 220  
 helicopter control matrix, 211  
 helicopter flight control, 207  
 Heneghan method, 116-117, 121-124  
 heuristic search, 258  
 hierarchical approaches, 261  
 hierarchical architecture, 185  
 hierarchical fuzzy processors, 261  
 high dimensional systems, 191  
 high stepping rates, 84  
 hit-miss topology, 119  
 home position, 96  
 horizontal tail plane, 209  
 horizontal tracker, 90  
 hostile, 223  
 human domain experts, 255  
 human visual system, 101  
 hybrid system framework, 295  
 hyperbolic tangent function, 195  
 hyperplane, 192-193, 196  
 hysteresis thresholding, 116-117
- I**
- IF-THEN rule, 27-28  
 image binarization, 106  
 image *complexity*, 104  
 image fuzzification function, 111  
 image segmentation, 124  
 image-expert, 122-123  
 indicator function, 121  
 inert, 223  
 inertia frame, 238  
 inference decision methods, 317  
 inferential conclusion, 317  
 inferential decision, 317  
 injection molding process, 315  
 inner loop controller, 87  
 integral time absolute error, 54  
 inter-class similarity, 252  
 internal dependencies, 169
- interpolation property, 203  
 interpolative nature, 262  
 intersection, 20, 23-24, 31, 180  
 interval sets, 178  
 interval type-2 FLC, 181  
 interval type-2 fuzzy sets, 177, 180-181, 184  
 inter-vehicle gap, 135  
 intra-class similarity, 252  
 inverse dynamics control, 228, 230  
 inverse dynamics method, 227  
 inverse kinematic, 299
- J - K**
- join, 180  
 Kalman gain, 213  
 kinematic model, 299  
 kinematic modeling, 299-300  
 knowledge based gear position decision, 148, 153  
 knowledge reasoning layer, 132  
 knowledge representation, 250  
 knowledge-based GPD model, 146
- L**
- labyrinths, 169  
 laser interferometer transducer, 83  
 laser tracker, 301  
 laser tracking system, 53, 63, 65, 75, 78-79, 83-85, 87, 98, 301  
 lateral control, 131, 138  
 lateral cyclic pitch angle, 209  
 lateral flapping angle, 210  
 leader, 238-239  
 linear control surface, 55  
 linear fuzzy PI, 61  
 linear hover model, 213  
 linear interpolation, 300-301, 306-307, 309, 313  
 linear interpolation method, 309  
 linear optimal controller, 207, 217  
 linear P controller, 73  
 linear state feedback controller, 7  
 linear structures, 117  
 linear switching line, 198  
 linear time-series models, 283  
 linguistic variables, 18, 25, 27, 90, 102, 175, 208, 258

- load shedding, 261  
 load-following capabilities, 288, 297  
 loading dock, 159-161, 170, 172  
 longitudinal control, 130-132  
 longitudinal cyclic pitch angle, 209  
 longitudinal flapping angle, 210  
 lookup table, 18, 31-35, 40, 44, 46, 47-48, 51, 65, 70, 74, 93, 300, 302, 304-305  
 lower membership functions, 179-180  
 LQ feedback gains, 208  
 LQ linear controller, 208  
 LQ optimal controller, 208  
 LQ regulator, 208  
 L-R fuzzy numbers, 121  
 Luenburger observer, 6  
 Lyapunov function, 5, 192, 284
- M**  
 Mamdani model, 40, 46  
 Mamdani's method, 242  
 Mamdani-type controller, 208  
 maneuverability, 164, 207, 209, 288  
 manual transmissions, 145  
 mapping function, 102, 104  
 marginal distribution functions, 259  
 market-basket analysis, 251-252  
 massive databases, 249  
 matched filtering, 115  
 mathematical morphology, 117, 127  
 mating pool, 59-60  
 max member principle, 106  
 max-dot method, 40-41, 46  
 mean distance function, 119  
 mean max membership, 106  
 mean of maximum method, 29  
 mean set, 118-121  
 measuring beam, 86  
 mechanical coupling effects, 87  
 mechanical layer, 132  
 median filter, 105  
 meet, 7, 50, 139, 180, 183, 302  
 membership degree, 39, 257  
 membership functions, 18, 25, 81  
 membership mapping processes, 56  
 miniature acrobatic helicopter, 208  
 minor steady state errors, 217  
 mixed-fuzzy controller, 92  
 mobile robot control, 130, 175, 181  
 mobile robots, 171, 175-176, 183, 187-189  
 model predictive control, 280, 287  
 model-based control, 224  
 modeless compensation, 300  
 modeless robot calibration, 299-301, 312-313  
 modern combined-cycle power plant, 279  
 modular structure, 172  
 mold-design optimization, 323  
 mold-design process, 323  
 molded part, 318-321, 323  
 morphological methods, 115  
 motor angular acceleration, 3  
 motor plant, 3  
 motor speed control, 2  
 moving average filter, 105  
 multilayer fuzzy logic control, 276  
 multimachine power system, 262  
 multivariable control, 280  
 multivariable fuzzy PID control, 285  
 multivariable self-tuning controller, 283, 295  
 mutation, 59  
 mutation probability, 59-60  
 mutual interference, 88
- N**  
 navigation control, 160  
 neural fuzzy control, 19, 36  
 neural networks, 173, 237, 255, 280, 284, 323  
 neuro-fuzzy control, 237  
 nominal plant, 2-4  
 nonlinear adaptive control, 237  
 non-linear control, 2, 159  
 nonlinear mapping, 55  
 nonlinear switching curve, 198-199  
 nonlinear switching function, 200  
 nonvolatile memory, 44  
 normalized universe, 266
- O**  
 objective function, 59, 74-75, 77, 107, 281-282, 284, 287, 289-291,

- 295
- obstacle avoidance, 166, 169, 187-188, 223-225, 227, 231
- obstacle avoidance behaviour, 187-188
- obstacle sensor, 224, 228
- off-line defuzzification, 34
- off-line fuzzy inference system, 302, 304
- off-line fuzzy technology, 300
- off-line lookup tables, 302
- offsprings, 59-60
- on-line dynamic fuzzy inference system, 302
- online tuning, 203
- open water trial, 202
- operating point, 210
- optical platform, 92
- optimal control table, 300
- optimal feedback gain, 208, 215-216
- optimal gains, 207
- original domain, 102
- outer loop controller, 85, 87
- outlier analysis, 251, 253
- output control gains, 92
- overshoot, 3-4, 6-7, 60-61, 75-76, 94, 96, 193, 229, 266
- P**
- path tracking, 223, 232-234
- pattern evaluation, 250
- pattern vector, 150-151
- PD controller, 4, 54-55, 68-69, 71, 74, 76-77, 79, 134, 163, 165, 202
- perception domain, 102
- performance index, 60, 207
- perturbed plants, 3, 7
- phase margins, 207
- phase-plan mapping fuzzy control, 19
- photovoltaic power systems, 261
- phugoid mode, 212
- PID, 1-4, 8, 13, 19, 53, 61, 64-65, 74, 80, 84-85, 87-90, 92-98, 192
- PID-fuzzy control, 19
- piecewise nonlinear surface, 193
- pitch angle, 202, 209, 217
- pitch controller, 193, 201-202
- pitch error, 193, 201
- pitch error rate, 193, 201
- pitch subsidence, 212
- planetary gearbox, 145
- point-in-time transaction, 252
- polarizing beam-splitter, 86
- poles, 4, 94, 96
- position sensor detectors, 84
- positive definite matrix, 213
- post fault, 268, 270
- post-fault trajectory, 273
- pre-defined membership functions, 302
- prediction, 251, 258, 281-283, 287, 290
- predictive control, 280, 282-287, 290-291, 293-297
- predictive supervisory controller, 284
- preview distance control, 129
- principal regulation level, 279
- probabilistic reasoning approach, 259
- probability space, 118
- Problem understanding phases, 254
- production rules, 316
- pursuer car, 136, 138-140
- pursuer vehicle, 136, 138, 140
- Q**
- quadrant detector, 79, 92
- quadrant photo detector, 85
- quadratic optimal technology, 208
- quadrilateral obstacle, 231
- R**
- radial basis function, 284
- random closed set, 118
- random compact set, 118-120
- rapid environment assessment, 191
- reference beam, 86
- relative frame, 240
- relay control, 195
- release distance, 169
- residual forces, 217
- retinal vessel detection, 115, 117
- RGB band, 115
- Riccati equation, 207, 213-214

rise time, 3, 54, 60-61, 75-76  
 road-environment estimator, 148  
 robot kinematics, 299  
 robot workspace, 299-302, 309  
 robust control, 2, 84, 280  
 robust controller, 2, 8, 90  
 robust fuzzy controller, 2, 7  
 robustness property, 5, 203  
 roll subsidence, 212  
 rotor blade flap angle, 209  
 rotor blades, 210  
 rudder, 193, 201  
 rule base size, 191, 199-200  
 rule output function, 191, 193, 198-199, 203  
 Runge-Kutta method, 61

**S**

sampling period, 96  
 saturation function, 195, 199  
 saturation functions, 162  
 scaling factor, 54, 72-73  
 scaling gains, 67, 69  
 S-curve waveform, 25  
 secondary membership function, 178  
 secondary memberships, 179, 181  
 selection, 59  
 self-learning neural network, 159  
 self-organizing fuzzy control, 261  
 self-tuning adaptive control, 280  
 self-tuning control, 191  
 semi-positive definite matrix, 213  
 sensitivity indices, 177  
 sequence-based analysis, 251-252  
 sequential quadratic programming, 283, 292  
 sets type-reduction, 184  
 setting time, 54, 60-61  
 settling time, 75-76, 94, 96  
 SGA, 59  
 shift points, 152  
 shift schedule algorithms, 148  
 shift schedules, 152, 156  
 shifting control, 145, 147  
 shifting schedules, 146, 152  
 shift-schedule tables, 152  
 sideslip angle, 210  
 sigmoidal waveform, 25

sign function, 195, 199  
 simplex optimal algorithm, 80  
 single gimbal system, 96  
 single point mass obstacle, 223  
 singleton fuzzification, 181-182  
 sinusoidal waveform, 94, 300, 309  
 sliding function, 192  
 sliding mode control, 1-2, 4, 8, 191, 193, 195-196, 203  
 sliding mode fuzzy controller, 193, 198-200  
 sliding mode fuzzy heading controller, 201  
 sliding pressure control, 280  
 sliding region, 192, 201  
 sliding surface, 5-6, 192-193, 195-198, 200  
 sliding-mode fuzzy control, 19  
 soft constraints, 281, 287  
 space-gap, 135  
 special-purpose processors, 48  
 spectral mapping theorem, 216  
 speed adaptation, 138  
 speed control, 2, 84, 130-131, 133, 160  
 spiral subsidence, 212  
 sporadic alternations, 257  
 state feedback controller, 213  
 state transition, 167-169  
 state transition matrix, 216  
 state-weighting matrix, 207  
 static fuzzy logic controller, 43  
 static MIMO system, 243  
 steady state error, 4, 54, 79, 90, 94, 96, 98, 192  
 steam turbine, 279  
 steam valving, 261  
 step response, 4, 7, 53, 76, 91, 193, 219  
 stern plane, 193, 201  
 sup operation, 183  
 supervisory control, 191, 280, 289  
 supervisory layer, 262-264, 277  
 support function, 118  
 support of a fuzzy set, 26  
 sup-star composition, 182-183  
 surviving solutions, 257

- swing curves, 271, 274-275  
 switching band, 198  
 switching curve, 198, 200  
 switching function, 191, 194, 196-198, 200  
 switching variable, 228  
 system trajectory, 192, 195
- T**
- tail plane, 210  
 tail rotor, 209-210  
 tail rotor derivation, 210  
 Takagi-Sugeno fuzzy methodology, 287  
 target displacement, 87  
 target time gap, 136  
 t-conorm maximum, 132  
 thermocouple sensor fault, 289  
 thickness variable, 319-320  
 three-beam laser tracker, 85  
 three-gimbal system, 96  
 throttle pressure, 134  
 throttle-opening degree, 149  
 thyristor control, 261  
 time delay, 63, 75, 91, 93-94, 281  
 time optimal robust control, 203  
 time-gap, 135-137, 139-140  
 time-gap derivative, 136  
 time-gap error, 136  
 time-invariant fuzzy system, 215  
 t-norm minimum, 132  
 torque converter, 145  
 tracking error, 79, 84-85, 92, 244  
 tracking gimbals, 87  
 tracking mirror, 85, 87  
 tracking performance, 84-85, 88, 90, 192  
 tracking speed, 75, 79, 83-84, 88, 90, 92, 97, 287  
 trajectory mapping unit, 161, 172  
 transfer function, 2-5, 61-63  
 transient response, 92, 193  
 transient stability, 261, 268, 270, 275-276  
 transient stability control, 268  
 trapezoidal waveform, 25  
 triangular fuzzy set, 319  
 triangular waveform, 25
- trim, 208, 210-211, 213, 217, 220, 237  
 trimmed points, 210  
 TS fuzzy gain scheduler, 217  
 TS fuzzy model, 207, 290  
 TS fuzzy system, 208, 215, 217, 220  
 TS gain scheduler, 217  
 TS model, 207, 287  
 TSK model, 40-41, 46  
 TS-type controller, 208  
 tuning function, 70, 72  
 turbine following mode, 280, 283  
 turn rate, 210  
 turning rate regulation, 208, 214, 217  
 two-DOF mirror gimbals, 87  
 two-layered FLC, 231  
 two-level hierarchy controllers, 275-276  
 two-module fuzzy logic control, 238  
 type-0 systems, 192  
 type-1 FLC, 176-177, 181-182, 185-188  
 type-1 fuzzy sets, 177-179, 181, 185, 187  
 type-1 membership functions, 176, 179, 183  
 type-2 FLC, 176-177, 180-183, 185-189  
 type-2 fuzzy set, 176-180  
 type-2 interval consequent sets, 184  
 type-2 membership function, 176-178  
 type-reduced set, 181, 183-185  
 type-reduction, 183-184
- U**
- UH-1H helicopter, 208  
 uncertain poles, 94, 96  
 uncertain system, 93-94, 96  
 uncertain zeros, 94, 96  
 underlying domain, 259  
 union, 20, 23-24, 30, 177, 180  
 unit control level, 279  
 universe of discourse, 19-24, 42, 57, 151, 153, 305  
 unmanned aerial vehicles, 223  
 unmanned helicopter, 208

unstructured dynamic environments, 177  
 unstructured environments, 175-177, 179, 185, 187, 189  
 upper membership function, 179

**V**

valve outlet pressure, 280  
 vapor pressure, 280  
 variable structure controller, 194, 204  
 velocity feedback, 87  
 vertical fin, 209  
 vertical tracker, 90  
 vertical tracking gimbal, 91  
 vessel detection, 115, 121-122, 124-125  
 vessel networks, 117  
 vessel segmentation, 115, 120  
 vessel tracking algorithms, 115  
 vision-driven robotics, 87  
 Vorob'ev fuzzy set average, 121-123  
 Vorob'ev mean, 118-120  
 vortex, 237

**W**

Wang and Mendel's algorithm, 257  
 WARP, 49  
 weak link, 270, 273  
 weighing factor, 305  
 weighting coefficients, 75  
 weighting function, 213  
 weld line, 315, 318-323  
 western states coordinating council, 269  
 Westinghouse turbine-generator, 283  
 wind-diesel power systems, 261  
 Wingman, 237-240, 246  
 wingman aircraft, 238-239  
 wingman velocity, 239

**Y-Z**

Yager operator, 292  
 Zana-Klein membership function, 124  
 Zana-Klein method, 116-117, 121, 123-124  
 zeros, 94, 96  
 $\mu$ -law function, 54  
 $\mu$ -law tuning method, 54