ON FUZZY LOGIC SYSTEMS, NONLINEAR SYSTEM IDENTIFICATION, AND ADAPTIVE CONTROL

by

JAMES X. LEE, M.ENG.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Department of Mechanical and Aerospace Engineering
Ottawa-Carleton Institute
for Mechanical and Aerospace Engineering

Carleton University
Ottawa, Ontario

Nov. 19, 1997

Abstract

A broad range of topics concerning fuzzy logic systems, nonlinear system identification and adaptive control are addressed in this thesis to achieve our main objective, which is to develop effective and reliable fuzzy logic approaches for identification and control of ill-defined, nonlinear dynamic systems.

First, improvements are made of existing fuzzy logic systems, which include defining two on-line quantitative measures for IF-THEN rule performance, and introducing a statistical confidence measure for approximation accuracy of FLS estimators. To facilitate on-line applications, a simplification is proposed of fuzzy inference computation, and the bounds of approximation errors are derived. Next, a complete procedure is presented for formulating expert knowledge based fuzzy logic controllers, and experimental results are demonstrated on a real mechanical system. The empirical FLC is also compared with PD controllers, and their respective properties discussed. Following is an optimal training scheme for fuzzy logic systems which combines a backpropagation algorithm with a least square estimation technique, synergistically combining them.

Observing the fact that the fuzzy logic systems being used to date are static in nature while the physical systems of interest are generally dynamic, a novel fuzzy logic system structure, the DFLS, which is characterized by inclusion of dynamics, is proposed, and its universal approximation property proved. Based on the DFLS, an identification algorithm is further developed, and its stability proper-

ties analysed theoretically. Its application to nonlinear, ill-defined dynamic systems is illustrated via a variety of examples, where the significance of human expert knowledge in improving system performance is demonstrated and a comparison of performance between DFLS and FLS identifiers is presented. In addition, a novel DFLS based indirect adaptive control scheme is developed, and its closed loop system performance and stability properties theoretically analysed. Two approaches are presented to estimate an unknown control gain function, g. One is based on a self-tuning scheme, the other is a FLS approach, and their respective properties are discussed. The DFLS adaptive control algorithm is applied to a variety of nonlinear systems, including a real mechanical system, and satisfactory results are observed in all situations, which demonstrates the effectiveness of the proposed control approach in dealing with nonlinear, ill-defined systems. Finally, a recurrent DFLS, the RDFLS, is introduced, its universal approximation property proved, and a RDFLS based stable identification algorithm developed. The stability properties of the RDFLS identifier are theoretically analysed, and its application to nonlinear systems is demonstrated via simulation examples.

Acknowledgements

I wish to extend my profound gratitude to my thesis supervisor, Dr. George Vukovich, for his invaluable guidance, generous suggestions, constant support and encouragement throughout the course of this research. His remarkable sharp insights into problems and his openness to the known and unknown world make discussions with him a inspiring and delightful experience.

I would like to thank Dr. Robert Bell, Chair of the Department of Mechanical and Aerospace Engineering, for his help and leadership in solving various difficulties in the course of this work.

I would like to express my special appreciation to Dr. Fred Afagh, my former graduate supervisor for Master's studies, for him to always stand beside me and ready to give his hands. I would also like to thank him for the informative discussions we had while in deriving the dynamic model of the flexible link manipulator in this work.

I would like to express my appreciation to the staff of Canadian Space Agency, where my experiments were conducted, for providing a pleasant and supportive environment, as well as technical helps in the course of this research. Especially, my friend, Dr. Yan-Ru Hu, offered numerous technical advises, and helped me in upgrading the personal computer used as the controller, solving timing problems in sampling process, and preparing photographs of the testbed.

I would like to thank staff of Department of Mechanical and Aerospace Engineering for providing an efficient working environment and various support in the course of this research.

I would like to thank my colleagues, Dr. Aria Alasty and Dr. Aghil Yousefi-Koma, for all the informative discussions we had during this research.

The financial support from Province of Ontario through Ontario Graduate Scholarships, Carleton University through graduate scholarships and teaching assistantships, Canadian Space Agency through research grants, are gratefully acknowledged, without which this work would have been impossible.

I take this opportunity to express my deepest appreciation to my wife Guojin for her encouragement, patience, tolerance, sacrifice, and support throughout the period which was devoted to this work. She was the one who absorbed the hardship, while carefully created a loving and comfortable environment for me and for our child, Victor.

Dedicated To

Guojin, Victor, and My Parents

Contents

Acceptance Form ii

Abstract iii

Acknowledgements v

Table of Contents viii

List of Tables xvi

List of Figures xvii

List of Symbols xxiii

1 Introduction 1

1.1 General Background 1

1.2 Research Objective 3

1.3 Thesis Outline 3

2 A Brief Overview of Modeling and Control of Nonlinear Systems 6

2.1 Conventional Methodologies 6

2.1.1 Control Techniques for Nonlinear Systems 6
2.1.2 Identification Approaches for Nonlinear Systems 8
2.2 Intelligent Techniques 10
2.2.1 Artificial Neural Networks 10
2.2.2 Fuzzy Sets and Fuzzy Logic 14
2.2.3 Remarks 25
2.3 Modeling and Control of Flexible Link Manipulators 27
2.3.1 Introduction 28
2.3.2 Dynamics of Flexible Link Manipulator 29
2.3.3 Position Control of Flexible Link Manipulators 32

3 Brief Review of Fuzzy Sets and Fuzzy Logic **35**
3.1 Introduction 35
3.2 Basics of Fuzzy Sets and Fuzzy Logic 37
3.2.1 Fuzzy Sets 37
3.2.2 Operations on Fuzzy Sets 38
3.2.3 Fuzzy Relations and Compositions 40
3.2.4 Linguistic Variables and Approximate Reasoning 41

4 Fuzzy Logic Systems **44**
4.1 Introduction 44
4.2 Fuzzification Interface 45
4.3 Fuzzy Rule Base 47
4.4 Fuzzy Inference Engine 49
4.5 Defuzzification Interface 49
4.6 Overall Mappings of Fuzzy Logic Systems 51

4.6.1 FLS-I 52
4.6.2 FLS-II 53
4.6.3 Membership Functions and an Overall FLS Mapping 55

5 More On Fuzzy Logic Systems **58**

5.1 Quantitative Measures of IF-THEN Rule Performance 58
5.1.1 Introduction 58
5.1.2 Quantitative Measures 59
5.1.3 Numerical Examples 61
5.1.4 Concluding Remarks 66
5.2 Statistical Confidence Measure of Fuzzy Logic System Performance . 67
5.2.1 Interval Estimation Problem 67
5.2.2 Confidence Interval for the Mean of FLS Approximation Errors 68
5.2.3 Confidence Interval for the Mean of Prediction Errors of Individual IF-THEN Rules 69
5.2.4 Numerical Examples 71
5.3 On Computational Issues of Fuzzy Logic Systems 78
5.3.1 Introduction 78
5.3.2 Approximation in FLS Computation 78
5.3.3 Bounds of Approximation Errors 80
5.3.4 Proof of Theorem 5.1 86

6 Fuzzy Logic Controller – An Application of the Theory of Fuzzy Sets and Fuzzy Logic **90**

6.1 Introduction 90
6.2 Basics of Fuzzy Logic Controller Design 91

6.3 Fuzzy Logic Control of Flexible Link Manipulator: Design and Experimental Demonstrations 94
6.3.1 Active Link Vibration Control with FLC 94
6.3.2 Vibration and Position Control with FLC 102
6.4 Experiments with PD Control 105
6.4.1 Introduction 105
6.4.2 Active Link Vibration Control with PD Controller 105
6.4.3 Vibration and Position Control with PD Controller 106
6.4.4 Remarks 110
6.5 Trajectory Control of a Flexible Link Manipulator - A Simulation Example with FLC 111
6.6 Concluding Remarks 115

7 Optimal Training For Fuzzy Logic Systems **116**
7.1 Preliminary 116
7.2 Optimal Training for FLS 118
7.2.1 Backpropagation Training 118
7.2.2 Least-Squares Estimation 119
7.2.3 LSE-BP Training of FLS 120
7.3 Numerical Examples 121
7.4 Concluding Remarks 127

8 The Dynamic Fuzzy Logic System and Nonlinear System Identification **128**
8.1 Introduction 128
8.2 The DFLS and Universal Approximation 129

8.3 DFLS Based Identification for Nonlinear Dynamic Systems and Stability Analysis . . . 134
8.3.1 Preliminary . . . 134
8.3.2 DFLS Identification Algorithm . . . 135
8.3.3 Proof of Theorem 8.2 . . . 139
8.3.4 Remarks . . . 143
8.4 Identification of Nonlinear Systems - Illustrative Applications of DFLS 146
8.4.1 Example 1 - A Nonlinear System with Chaotic Behavior . . . 146
8.4.2 Example 2 - Chaotic Glycolytic Oscillator . . . 155
8.4.3 Example 3 - A Flexible Link Manipulator with Nonlinear Joint Friction . . . 164
8.4.4 Expert Knowledge and Identifier Performance . . . 176
8.5 Experimental Demonstration - Identification of a Mechanical Manipulator . . . 178
8.6 Concluding Remarks . . . 185

9 On Adaptive Control of Nonlinear Systems – DFLS Approach 187

9.1 Introduction . . . 187
9.2 Preliminaries and Objective . . . 189
9.2.1 Dynamic System . . . 189
9.2.2 DFLS Identifier . . . 190
9.2.3 Objective . . . 192
9.3 Control Law of the System . . . 193
9.4 Adaptive Law for Parameter Vector $\overline{\mathbf{Y}}$ . . . 194
9.5 Adaptive Laws for $\hat{g}$ . . . 196

9.5.1 Self-Tuning Scheme for $\hat{g}$ 196
9.5.2 FLS Approach for $\hat{g}$ 198
9.6 Stability Properties of DFLS Control Algorithm 200
9.7 A Special Situation : $g(\mathbf{x})$ Known 201
9.8 Proof of Theorem and Corollaries 202
9.8.1 Proof of Theorem 9.1.1 202
9.8.2 Proof of Theorem 9.1.2 - Estimation of $g(\mathbf{x})$ with Self-tuning Scheme 205
9.8.3 Proof of Theorem 9.1.2 - Estimation of $g(\mathbf{x})$ with FLS 208
9.8.4 Proof of Theorem 9.1.3 212
9.8.5 Proof of Theorem 9.1.4 215
9.8.6 Proof of Theorem 9.1.5 - Estimation of $g(\mathbf{x})$ with Self-tuning Scheme 216
9.8.7 Proof of Theorem 9.1.5 - Estimation of $g(\mathbf{x})$ with FLS 220
9.8.8 Proof of Theorem 9.1.6 227
9.8.9 Proof of Corollaries 9.1 and 9.2 228
9.9 Illustrative Applications 229
9.9.1 Example 1 - Trajectory Control of a Simple Nonlinear System 229
9.9.2 Example 2 - Control of a Nonlinear System with Chaotic Behavior 242
9.10 Experimental Demonstration - Trajectory Control of a Mechanical Manipulator 253
9.11 Concluding Remarks 263

10 The Recurrent Dynamic Fuzzy Logic System and Nonlinear System

Identification **267**

10.1 Preliminary 267

10.2 Universal Approximation Property 268

10.3 RDFLS Identification of Nonlinear Systems 273

10.3.1 Preliminary 273

10.3.2 Adaptive Law and Stability Properties 274

10.3.3 Proof of Theorem 10.2 276

10.4 Illustrative Applications - Identification of Nonlinear Systems 281

10.4.1 Example 1 - Identification of a Simple Nonlinear System . . . 281

10.4.2 Example 2 - Duffing Forced-Oscillation System 283

10.5 Concluding Remarks 289

11 Conclusions and Suggestions for Future Work **290**

11.1 Conclusions 290

11.2 Summary of Major Contributions 296

11.3 Suggestions for Future Work 299

Bibliography **301**

Appendices **328**

A Dynamic Model of Flexible Single Link Manipulator **328**

A.1 System Configuration 328

A.2 Potential Energy and Kinetic Energy 332

A.3 Derivation of Equations of Motion 335

A.4 Remarks 342

A.5 Eigenvalues and Eigenfunctions 344

A.6 Orthogonality of Eigenfunctions 348
A.7 Uncoupled Equations of Motion of Undamped System 350
A.8 Damping Resistances 352
A.9 State Space Representation 354
A.10 The Joint Friction of the Manipulator 356

B Experimental Testbed – A Single Link Robotic Manipulator 359
B.1 System Configuration 359
B.2 Direct Drive Motor and Interface to Computer 360
B.2.1 Introduction 360
B.2.2 Motor Operating Modes 362
B.3 The Flexible Beam 363
B.4 Ultrasonic Position and Orientation Sensing System 363

C Thesis Subject Related Publications 371

List of Tables

5.1 IF-THEN Rules for Linguistic Variable u 62

5.2 Fuzzy Rule Base 81

5.3 Bounds of Approximation Errors 85

6.1 Fuzzy Rule Base of the FLC for Vibration Control Experiment 99

6.2 Fuzzy Rule Base of the FLC for Trajectory Control 113

7.1 Initial and Final Values of FLS parameters 124

8.1 Linguistic Description of Dynamic Behavior of $\dot{o}(0,t)$ 176

List of Figures

3.1 The Concept of Fuzzy Set 36

4.1 Basic Structure of Fuzzy Logic System 45

4.2 Illustration of Fuzzifiers 46

4.3 Gaussian and Triangular Membership Functions 56

5.1 Distribution of Fire-Density Indices 61

5.2 Completeness of Fuzzy Rule Base 63

5.3 Data Distribution, Fuzzy Partition and Fire-Density Index 65

5.4 Structure of FLS Approximator 71

5.5 FLS Outputs and Confidence Intervals for Expected Values of Approximation Errors 75

5.6 Confidence Intervals for Expected Values of Prediction Errors of Individual Rules 76

5.7 Confidence Intervals for Individual Rules at k=50, 100, and 200 . . . 77

5.8 A Fuzzy Partition 79

6.1 Reconfigured Fuzzification Interface 92

6.2 A Normalized Fuzzy Partition 93

6.3 Reconfigured Defuzzifier 94
6.4 Fuzzy Logic Controller Development 96
6.5 FLC for Vibration Control Experiment 97
6.6 Primary Fuzzy Sets for ϵ in Vibration Control Experiment with FLC 99
6.7 Illustration of Δ'^{i} and $\overline{\delta}^{i}$ 101
6.8 Free Vibration at the Tip of the Link When Disturbed 101
6.9 Active Vibration Control Experiment with FLC 102
6.10 Experiment on Vibration and Position Control with FLC 104
6.11 PD Control System For Vibration Control Experiment 106
6.12 Active Vibration Control Experiment with PD Controller 107
6.13 Experiment on Vibration and Position Control with PD Controller . . 109
6.14 Control System Diagram for Trajectory Control with FLC 112
6.15 Primary Fuzzy Sets for ϵ in Trajectory Control Example with FLC . 113
6.16 Trajectory Control of a Flexible Link Manipulator 114

7.1 Training of the FLS Identifiers 125
7.2 Testing of the Trained FLS Identifiers 126

8.1 Dynamic Fuzzy Logic System 133
8.2 Phase Plane Trajectory - Example 1 147
8.3 Primary Fuzzy Sets in $\mathcal{X}_1$ and $\mathcal{X}_2$ for D_{x_1} 148
8.4 Identification of x_1 - Example 1 150
8.5 Primary Fuzzy Sets in $\mathcal{X}_1$ and $\mathcal{X}_2$ for D_{x_2} 152
8.6 Identification of x_2 - Example 1 153
8.7 System Dynamics for Different Initial Conditions 154
8.8 Test of Trained Identifiers for $\mathbf{x}(0) = \{2.\ 3\}^T$ 155

8.9 Phase Plane Trajectory – Example 2 157

8.10 Identification of x_1 with $\mathbf{x}(0) = \{1.5,\ 1.5\}^T$ – Example 2 158

8.11 Identification of x_2 with $\mathbf{x}(0) = \{1.5,\ 1.5\}^T$ – Example 2 160

8.12 Identification of x_1 and x_2 Using FLS ("First-Type Identifier") . . . 161

8.13 Identification of x_1 and x_2 Using FLS ("Second-Type Identifier") . . 162

8.14 Identification of x_1 and x_2 Using DFLS (For Comparison with FLS Results) 163

8.15 Inputs-Outputs of DFLS identifiers – Example 3 166

8.16 Command Voltage $V_m = 0.5\sin(t)$ and Torque T_h – Example 3 169

8.17 Identification of Angular Displacement and Velocity for $V_m = 0.5\sin(t)$ – Example 3 171

8.18 Identification of Vibrational Displacement and Velocity at the Tip of the Beam for $V_m = 0.5\sin(t)$ – Example 3 172

8.19 Command Voltage V_m and Torque T_h for Test of Trained DFLS Identifier – Example 3 173

8.20 Test of Trained Identifiers for Angular Displacement and Velocity of the Motor – Example 3 174

8.21 Test of Trained Identifiers for Vibrational Displacement and Velocity of the Tip of the Beam – Example 3 175

8.22 Training of the Identifier for Angular Velocity of the Motor, $\dot{o}(0,t)$, with and without Expert Knowledge Incorporated 177

8.23 Inputs-Outputs of DFLS identifiers – Identification Experiment . . . 178

8.24 Training Process for Command Input $V_m = 1.2\sin(0.2\pi t)$ – Identification Experiment 181

8.25 Training Process for Command Input $V_m = 2\sin(0.6\pi t)$ - Identification Experiment 182
8.26 Training Process for Complex Command Input - Identification Experiment 183
8.27 Test of Identifiers for Complex Command Input - Identification Experiment 184

9.1 Trajectory Control for $M_e \triangleq 15$ — Example 1 (Function g Known) . 235
9.2 Trajectory Control for $M_e \triangleq 6$ — Example 1 (Function g Known) . . 236
9.3 Trajectory Errors for Different Error Bounds — Example 1 237
9.4 Trajectory Control — Example 1 (Function g Unknown and Estimated with Self-tuning Scheme) 238
9.5 Estimation of x and $g(x)$ — Example 1 (Function g Unknown and Estimated with Self-tuning Scheme) 239
9.6 Trajectory Control — Example 1 (Function g Unknown and Estimated with FLS) 240
9.7 Estimation of x and $g(x)$ — Example 1 (Function g Unknown and Estimated with FLS) 241
9.8 Trajectory Control — Example 2 (Function g Known) 247
9.9 Phase Plane Expression of Controlled Trajectory — Example 2 (Function g Known) 248
9.10 Trajectory Control — Example 2 (Function g Unknown and Estimated with Self-tuning Scheme) 249
9.11 Estimation of x_2 and $g(\mathbf{x})$ — Example 2 (Function g Unknown and Estimated with Self-tuning Scheme) 250

9.12 Trajectory Control — Example 2 (Function g Unknown and Estimated with FLS) . . . 251
9.13 Estimation of x_2 and $g(\mathbf{x})$ — Example 2 (Function g Unknown and Estimated with FLS) . . . 252
9.14 Illustration of a Single Link Manipulator . . . 253
9.15 Trajectory Control of a Mechanical Manipulator — Experiment 1 . . . 259
9.16 Estimation of $\dot{w}$ and g — Experiment 1 . . . 260
9.17 Trajectory Control of a Mechanical Manipulator — Experiment 2 . . . 261
9.18 Estimation of $\dot{w}$ and g — Experiment 2 . . . 262

10.1 Concept of RDFLS . . . 268
10.2 Identification of x with RDFLS — Example 1 . . . 284
10.3 State Space Trajectory — Example 2 . . . 286
10.4 Identification of State Variables with RDFLS — Example 2 . . . 288

A.1 An Experimental Single Link Robotic Manipulator . . . 329
A.2 A Schematic Illustration of Single Link Robotic Manipulator . . . 330
A.3 Cross-Sectional Dimensions of the Beam . . . 334
A.4 Free Body Diagram of the Beam . . . 339
A.5 Shear Force on the Free End of the Beam . . . 343
A.6 First Three Flexural Eigenvalues and Eigenfunctions . . . 348
A.7 Dynamic Model of a Single Link Manipulator . . . 358

B.1 Experimental Testbed — A Flexible Link Mechanical Manipulator . . . 365
B.2 Schematic Illustration of the Experimental Testbed . . . 366
B.3 Internal Control Loop of the Motor . . . 366

B.4 Interface Between Motor Controller and Computer 367
B.5 Dynamic Model of the Manipulator in Torque Control Mode 367
B.6 Dynamic Model of the Manipulator in Velocity Control Mode 368
B.7 Dimensions of the Beam 368
B.8 Ultrasonic Position and Orientation Sensing System 369
B.9 Ultrasound Transmitter and Receiver 370

List of Symbols

Abbreviations

ANN	Artificial Neural Network
BP	Backpropagation
CM	Centroid Defuzzification Method
CI	Completeness Index (Chapter 5)
DFLS	Dynamic Fuzzy Logic System
FDI	Fire Density Index (Chapter 5)
FEM	Finite Element Method
FLS	Fuzzy Logic System
FLC	Fuzzy Logic Controller
GMP	Generalized Modus Ponens
LMM	Least-Maximum Defuzzification Method
LQG	Linear Quadratic Gaussian
LSE	Least-Square Estimation
MOM	Mean of Maximum Defuzzification Method
PDE	Partial Differential Equation
PD	Proportional-Derivative

PID	Proportional-Integral-Derivative
RDFLS	Recurrent Dynamic Fuzzy Logic System
ST	Self-Tuning

English Symbols

A_p^i, or B_p^i	primary fuzzy set in pth universe of discourse and appearing in ith IF-THEN rule
$A_{p j_p}$, or $B_{p j_p}$	the j_pth primary fuzzy set in pth universe of discourse
E^U, E^L	upper and lower confidence limits (Section 5.2)
$\hat{g}$	estimates of function $g(\mathbf{x})$, eq.(9.40)
$\mathbf{H}$	a constant positive definite symmetric matrix
L	length of a beam
M_{r_k}	upper bound of the absolute value of the static modeling error, r_k
$M_{\overline{\mathbf{Y}}_k}$	upper bound of the norm of the parameter vector $\overline{\mathbf{Y}}_k$
r_k	static modeling error, eq.(8.27)
R	set of real numbers
$t_{(1-\frac{\delta}{2})}$	$1-\frac{\delta}{2}$ percentile of a Student-t distribution (Section 5.2)
u, v, w, x, y, z	linguistic variables
$\mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}$	universes of discourse
U, V, W, X, Y, Z	fuzzy sets
$v(L,t)$	vibrational deflection at the tip of a beam
V_m	command voltage of the motor
$\mathbf{w}$	a vector with elements being linguistic variables, w_p, $p = 1, \ldots, P$

$w(L,t)$	displacement at the tip of a beam
$\mathbf{x}$	state vector
$\overline{y}^i$	center of fuzzy set Y^i (otherwise as specified)
$\overline{\mathbf{Y}}$	a vector with elements being $\overline{y}^i$
$\overline{\mathbf{Y}}^*$	the optimal parameter vector, eq.(8.30)
$\mathbf{z}$	input vector of a fuzzy logic system
$\overline{z}_p^i$	center of a fuzzy set Z_p^i

Greek Symbols

Δ_g	difference between $\hat{g}$ and g, eq.(9.27)
$\boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k}$	difference between $\overline{\mathbf{Y}}_k$ and $\overline{\mathbf{Y}}_k^*$, eq.(8.33)
γ	confidence coefficient (Section 5.2)
$\mu_{U^i}(u)$	grade of membership of u in U^i (for u being a crisp value), or, the membership function of U^i (for u being a variable)
$\phi(0,t)$	angular displacement at the root of the beam
$\psi_j(\mathbf{x})$	fuzzy basis function, eq.(9.42)
$\boldsymbol{\Psi}(\mathbf{x})$	a vector with elements being $\psi_j(\mathbf{x})$, eq.(9.41)
σ_p^i,	shape parameter for primary fuzzy set A_p^i
$\theta_i(\mathbf{z})$	a fuzzy basis function
$\boldsymbol{\Theta}$	a vector with elements being fuzzy basis functions

Signs

$\triangleq$	defined as
$\in$	in the set
$\notin$	not in the set
$\exists$	there exists
$\forall$	for any
$\subset$	subset
$\star$	t-norm (Chapter 3)
$\dot{+}$	s-norm (Chapter 3)
$\circ$	fuzzy composition
$\rightarrow$	fuzzy implication
$\equiv$	equivalent
$\cup$	union
$\cap$	intersection
f^{-1}	inverse of function f
$\dot{x}$	$\frac{dx}{dt}$
arg	argument
exp	exponential function
$\lambda_{max}(\mathbf{Q})$	the maximum eigenvalue of matrix $\mathbf{Q}$
$\lambda_{min}(\mathbf{Q})$	the minimum eigenvalue of matrix $\mathbf{Q}$
max	maximum
min	minimum
sup	supremum

Chapter 1

Introduction

1.1 General Background

Evolution in the regime of automatic control has been fueled by three major needs [5, 6]: the need to deal with increasingly complex systems, the need to satisfy increasingly demanding design requirements, and the need to attain these requirements with less precise advanced knowledge of the plant and its environment.

Control of well-defined linear systems is a mature subject with a variety of powerful methods and a long history of successful industrial applications [66, 120]. In contrast to this, constructive procedures similar to those available for linear systems do not exist for nonlinear systems [115]. Even so, the development and application of control methodologies for nonlinear systems have their own merits which are not undermined by the existence of these well developed methodologies for linear systems [151]. For example, linear control methods rely on the key assumption of small range operation for the linear model to be valid. When the required operation range is large, a linear controller is likely to perform very poorly or to be unstable, because the nonlinearities in the system can not be properly compensated for. Nonlinear con-

trollers, on the other hand, may handle the nonlinearities in large range operation directly, and result in improved control systems. Also, there are many nonlinearities whose discontinuous nature does not allow linear approximation. These so-called *hard nonlinearities* include Coulomb friction, saturation, dead-zones, backlash, and hysteresis, and are often found in control engineering. Their effects can not be dealt with by linear methods, and nonlinear analysis techniques must be developed to predict a system's performance in the presence of these inherent nonlinearities which frequently cause undesirable behavior of the control systems, such as instabilities or spurious limit cycles.

Fundamental to system analysis and control is the characterization of the system of interest by mathematical expressions, or so called system modeling. System models can be obtained either by manipulating mathematical descriptions of physical laws governing a particular system or by extracting system mapping information from experimental input-output data sets of the system. The former approach is generally referred to as *analytical modeling*, the latter as *identification*. With ill-defined, nonlinear systems, the effectiveness of the analytical modeling approach is severely limited and consequently, identification approaches have been developed.

Among various active research areas in the regime of identification and control of ill-defined, nonlinear systems, the fuzzy logic approach distinguished itself by its impressive successes in wide spectrum of practical applications, and by its capability to systematically incorporate human linguistic information into control and identification system design. Nevertheless, the fuzzy logic approach still faces many challenges. It has even not been viewed as a rigorous science by some people [126, 127] due to the lack of formal synthesis techniques in system design which guarantee the basic requirements of global stability and acceptable performance. The design of fuzzy logic systems has been ad hoc and their parameters often must be manually adjusted on a trial and error basis. It is the purpose of this work to ad-

dress some of the challenges in the regime of identification and control of ill-defined, nonlinear systems using the fuzzy logic approach.

1.2 Research Objective

The objective of this research is to develop effective fuzzy logic approaches for identification and control of nonlinear, ill-defined dynamical systems. This includes investigation of current fuzzy logic systems, identifying their weaknesses, making necessary improvements, and developing fuzzy logic based identification and control approaches that can be more effective and reliable in dealing with nonlinear, ill-defined dynamic systems to achieve better performance.

1.3 Thesis Outline

An overview of the issues related to this research is introduced in next chapter. A brief theory of fuzzy sets and fuzzy logic is presented in chapter 3. Following this, in chapter 4, is a detailed introduction of fuzzy logic systems (FLS) that is directly related to our later development.

In chapter 5, improvements to current fuzzy logic systems are presented. First, two simple quantitative measures are defined and integrated into fuzzy logic systems, which can indicate certain difficulties with IF-THEN rules and increase the reliability of the corresponding fuzzy logic systems. Next, a statistical confidence measure, the confidence interval, is introduced for fuzzy logic systems used in functional approximation problems, and is used to indicate statistical confidence of the approximation accuracy of both overall FLS output and prediction of individual IF-THEN rules. Following is a simplification of fuzzy inference computation for an important type of FLS. This results in a significant increase of calculation speed,

which is very important for on-line control applications. The bounds of approximation errors are derived in closed form and their values also tabulated for applications of potential practical interest.

Chapter 6 investigates the application of the theory of fuzzy sets and fuzzy logic to automatic control, where a complete procedure to formulate expert knowledge-based fuzzy logic controllers (FLC) is presented. It is demonstrated that human knowledge can be effectively incorporated into fuzzy logic systems to produce practical controllers. In addition, the mathematically formulated fuzzy logic controllers are amenable to treatment by powerful analytical tools and can be subsequently improved and analysed theoretically, by using the approaches developed in the following chapters. The empirical FLC and PD (proportional-derivative) control approaches are compared and experimental results are included.

Complementing the selection of fuzzy logic system parameters by expert knowledge or trial and error as in the cases of chapter 6, an optimal fuzzy logic system training approach is introduced in chapter 7 which combines a backpropagation (BP) training algorithm with a least squares estimation technique. The resulting optimal LSE-BP training scheme avoids the weaknesses of both LSE and BP approaches while combining the strength of both.

Following observation of the fact that the fuzzy logic systems being used to date are static in nature while the physical systems of interest are generally dynamic, a novel fuzzy logic system structure, the DFLS, which is characterized by inclusion of dynamics, is proposed in chapter 8, and its universal approximation property proved. Based on the DFLS, an identification algorithm is further developed and its stability properties analysed theoretically. Its application to nonlinear, ill-defined dynamic systems is illustrated via a variety of examples in which the significance of human expert knowledge in improving system performance is made clear, and comparisons of performance between DFLS and FLS identifiers are also demonstrated.

Following this in chapter 9, a novel DFLS based indirect adaptive control scheme is developed, and its closed loop system performance and stability properties theoretically analysed. Two approaches are presented to estimate an unknown control gain function, g: one is based on a self-tuning scheme, the other is a FLS approach. The DFLS adaptive control algorithm is applied to a variety of nonlinear systems including a real mechanical system, and satisfactory results are observed in all situations, which demonstrates the effectiveness of the proposed control approach in dealing with nonlinear, ill-defined systems. The potential advantages of the DFLS based controllers over static FLS based controllers are also discussed.

A recurrent DFLS, the RDFLS, is introduced in chapter 10, and its universal approximation property proved. Further, a RDFLS based stable identification algorithm is developed, its stability properties theoretically analysed, and its application to nonlinear systems demonstrated via simulation examples.

This work is concluded in chapter 11, where the major contributions of this research are highlighted, and possible topics as a continuation of this work are suggested.

A dynamic model of a flexible single link robotic manipulator is developed in appendix A. Although presented as an appendix, it is an integral portion of this research because it provides not only an important vehicle for simulation and experimental demonstration of the theory, but the derivation process itself also provides much insight into the mathematical model and provides clues for the formulation of the research plan in the early stages of this work.

The experimental setup, which is a flexible single link robotic manipulator, is described in appendix B.

A list of publications resulting from this research and up to the date of preparation of this manuscript is presented in appendix C.

Chapter 2

A Brief Overview of Modeling and Control of Nonlinear Systems

2.1 Conventional Methodologies

2.1.1 Control Techniques for Nonlinear Systems

In the analysis and control of nonlinear systems, although there are currently no mature, general methods for obtaining reliable controllers, there is, nevertheless, a rich collection of alternative and complementary techniques, each best applicable to particular classes of nonlinear control problems [59, 119, 151]. These techniques may be loosely classified into two groups. One includes conventional techniques such as trial and error, feedback linearization, robust control, adaptive control, and gain scheduling. The other covers the techniques in the realm of intelligent control systems, such as artificial neural network approaches, fuzzy logic approaches, genetic algorithm methods, and so on, [130, 132, 187, 198]. In this section, we briefly review the basic conventional control techniques; the other type of control methodologies is reviewed in later sections.

For the first group suggested above, Slotine and Li [151] presented a comprehensive, easy to follow and up to date tutorial on various methods, which the interested

reader can consult for more details. The brief introduction to major nonlinear control techniques we give here is primarily based on [151].

The idea of the trial and error approach is to use analysis tools, such as the phase plane method, the describing function method, and Lyapunov analysis to guide the search for a controller which can then be supported by analysis and simulations. Experience and intuition are critical in this process. However, for complex systems, trial and error often fails.

The feedback linearization approach first transforms a nonlinear system into a (full or partial) linear system, and then uses the powerful linear design techniques to complete the control design. Although successful in solving a number of practical nonlinear control problems, it is not applicable to non-minimum phase systems and does not guarantee robustness in the face of parameter uncertainty or disturbances.

The idea of robust nonlinear control is to design controllers based on the consideration of both the nominal model of the plant and some characterization of the model uncertainties (such as the knowledge that the load to be picked up by a robot is between 2 kg and 10 kg). This method has proven very effective in some practical control problems, but is applicable to primarily specific classes of nonlinear systems and is subject to weaknesses such as chattering and requirements for large control authority.

Adaptive control is an approach to deal with uncertain or time-varying systems. Although the term "adaptive" is broad in meaning, current adaptive control designs apply mainly to systems with known dynamic structure but unknown constant or slowly varying parameters.

Gain scheduling is a technique in which the well developed linear control methodology is applied to the control of nonlinear systems. The idea is to select a number of operating points which cover the range of system operation, and at each of these, the designer makes a linear time invariant approximation to the plant dynamics and

designs a linear controller for each linearized plant. Between operating points, the parameters of the compensators are interpolated, or scheduled, thus resulting in a global compensator. This method is conceptually simple, and also practically successful for a number of applications. But it has only limited theoretical guarantees of stability in nonlinear operation, and its computational burden is excessive, due to the necessity of computing many linear controllers.

2.1.2 Identification Approaches for Nonlinear Systems

System modeling is fundamental to system analysis and high performance controller design[1].

System modeling has two basic paradigms, analytical modeling and nonanalytical modeling. In analytical modeling, a system model is obtained from equations derived from physics, e.g., using Kirchhoff's laws to obtain the dynamic model of an electrical system, using Newton's second law for a mechanical system, and so on. However, it is well known that analytical modeling can be severely hampered by complexity of the physical systems to be modeled, or by the physical processes underlying a phenomenon being unavailable or even unknown, rendering analytical modeling impossible. The complexity may be due to inherent nonlinear and non-stationary aspects, and is difficult to handle effectively with current controller design and stability analysis techniques. To address these difficulties, identification approaches have been developed.

A vast literature exists on the characterization of nonlinear functionals. Some excellent tutorial surveys are available for reference [8, 14, 46, 106].

The most systematic approach to nonlinear system identification is that of Wiener [191, 192], which involves Laguerre and Hermite series expansions to model the dy-

[1]There are some excellent comments presented in [114, section 1.4] about the relationship between system modeling and control, which the interested reader may refer to.

namics and the nonlinear aspects, respectively. The model outputs are formed as an infinite series of products of Hermite polynomials in the Laguerre coefficients of the past of the input. A problem with this approach is the large number of coefficients to be estimated. Actually, this is such an important restriction that the method is rarely implemented [65].

A nonlinear system representation with potential for time domain as well as frequency domain applications is provided by the Volterra kernel representation, which leads to multilinear extensions of the transfer function notions used in linear systems [79]. This method, however, is difficult to extend to systems with feedback, and is for this reason of limited value [65].

For system control purposes, two classes of identification techniques have been found particularly useful in practical applications.

One class of identification techniques assumes a system dynamic structure with unknown parameters, and then adjusts its parameters so as to minimize the errors between the physical process and the model outputs in some optimal sense. The most successful examples of this kind are in the class of model reference adaptive control and self-tuning regulators [7, 41, 114, 151]. However, it is also known that this kind of approach generally suffers from such weaknesses as the requirement for explicit *a priori* knowledge of model structure, and ineffectiveness in dealing with intrinsic system nonlinearities [65, 151].

Another class of popular identification techniques that has recently emerged is that with universal nonlinear approximation capabilities, such as Artificial Neural Networks (ANN) and Fuzzy Logic Systems (FLS). This class of approaches views a physical system as a black box and identifies unknown system mappings based only on input-output data sets, and thus neither requires *a priori* knowledge of system structure, nor is impaired by the presence of nonlinearities and unknown internal dynamics. Methodologies of this kind will be introduced in the following sections.

2.2 Intelligent Techniques

The concept of intelligent control was introduced close to two decades ago by Saridis [146], and is recently gaining in recognition and visibility in the control systems establishment [187, 198, 207]. According to Åström and McAvoy [187, Chapter 1]: *an intelligent control system has the ability to comprehend, reason, and learn about processes, disturbances, and operation conditions. Factors that have to be understood and learned are primary process characteristics like static and dynamic behavior, characteristics of disturbances, and equipment-operating practices.. It would be desirable if this knowledge was acquired and stored in such a way that it can be used and retrieved. It would also be desirable that a system can autonomously improve upon its performance as experience is gathered.*

Although it appears that current control systems have a long way to go before they can qualify for the title of "intelligent control systems", impressive progress has been made due to the emergence of such methodologies as fuzzy logic, artificial neural networks, genetic algorithms, and so on [110, 130, 132, 187, 198, 207]. In the following, we present a brief overview of these subjects.

2.2.1 Artificial Neural Networks

The idea of a neural network was originally conceived as an attempt to model the biophysiology of the brain, i.e., to understand and explain how the brain operates and functions. The goal was to create a model capable of human thought processes [4, 105]. Today, there are two different, not always disjoint, groups of people exploring neural networks. The first group is composed of biologists, physicists and scientific psychologists who work toward developing a neural model that accurately mimics the behavior of the brain. The second group consists of engineers who are concerned with how *artificial* neurons can be interconnected to form networks with

interesting and powerful computational capabilities [39, 108, 112]. This second group treats the biological models as a functional starting point for research, which is the approach of interest for control engineers.

Two classes of neural networks have received considerable attention in recent years. One is multilayer feedforward neural networks [32, 40, 52, 108, 121, 144, 185], the other is feedback, or recurrent, neural networks [49, 50, 76, 77, 78, 95]. In a feedforward network, the output of any given neuron can not be fed back to itself directly or indirectly, and so its present output does not influence future outputs. When an input pattern is presented to the input terminals of a network, the neurons in the first layer compute output values and pass these values onto the next layer. Each layer in turn receives input values from the previous layer, computes output values and passes these values on to the next layer. When the output values of the final layer are determined, the computation ends. This rule can be violated only during the training phase, when the output of a neuron can be used to adjust its weights, thus influencing future outputs of the neuron. In a feedback network, any given neuron is allowed to influence its own output directly through self-feedback or indirectly through the influence it has on other neurons from which it receives inputs. Classically, multilayer feedforward networks have proven extremely successful in pattern recognition problems [25, 43, 148, 190] while recurrent networks have been used in associative memories as well as for solution of optimization problems [49, 51, 140, 162].

The ANN possesses some important characteristics [39, 56, 112]. First, a number of results have been published showing that a multilayer feedforward neural network can approximate arbitrarily well a continuous function [32, 40, 52]. Second, ANN has a highly parallel structure which lends itself immediately to parallel implementation, while the basic processing element in a neural network has a very simple structure so that with these together, fast overall processing can be achieved for a neural

network. Third, ANN has learning and adaptation ability.

ANNs have either fixed weights or adaptable weights. Networks with adaptable weights use learning laws to adjust the values of the interconnection strengths. If the neural network is to use fixed weights, then the task to be accomplished must be well defined *a priori*. The weights are determined explicitly from the description of the problem. Adaptable weights are essential if it is not known *a priori* what the correct weights should be.

There are two types of learning: supervised and unsupervised. Supervised learning occurs when the network is supplied with both the input values and the correct output values, and the network adjusts its weights based upon the error of the computed output. Unsupervised learning occurs when the network is only provided with input values, and the network adjusts the interconnection strengths based solely on the input values and the current network output. Details of various learning algorithms may be found in [39, 108, 112].

In the regime of control engineering, if an ANN is used to model an unknown plant, it is an identifier in the conventional sense, and if it is used to generate control signals directly, it assumes the role of a controller. Both multilayer feedforward neural networks and recurrent neural networks have shown great potential in dealing with nonlinear systems [29, 30, 70, 71, 83, 110, 115, 116, 133, 138, 145, 155]. Among various types of neural networks and training approaches, the multilayer feedforward neural network equipped with a backpropagation training algorithm has been particularly enthusiastically received among researchers for being simple yet effective.

The backpropagation algorithm, which is gradient based learning in general, was developed by Werbos in 1974 [185] but was not much noticed until it was rediscovered and popularized by Rumelhart et al. in 1986 [144]. It is now probably the most widely used training strategy for artificial neural networks. This method

computes local errors from the global errors, and in turn uses the local errors for local adaptation by means of gradient search techniques.

The basic approaches reported in the literature for applying ANNs in control include supervised control, direct inverse control, neural adaptive control, back-propagation through time, and adaptive critic methods [110, 187].

In supervised control, ANNs are trained on a database that contains the *correct* control signals to use in sample situations, i.e., ANNs are trained to copy an existing controller [110, 187]. But this approach requires the existence of a database of sensor inputs and desired actions.

In direct inverse control, ANNs directly learn the mapping from desired trajectories (e.g. of a robot arm) to the control signals which yield these trajectories (e.g. joint angles) [44, 110, 187]. This approach was originally proposed by Widrow et al. (1978, 1986) [188, 189] for the adaptive control of linear systems, and was demonstrated to be applicable to the control of nonlinear systems by Psaltis et al. (1988) [135]. A major problem with inverse identification arises when many plant inputs produce the same output, i.e., when the plant's inverse is not well defined. In this case, the network will attempt to map the same network input to many different target responses.

In neural adaptive control, neural networks are used to identify the unknown nonlinear systems, and adaptive control schemes are synthesized based on the identified system models [70, 71, 110, 115, 116, 187].

In backpropagation through time, the user specifies a utility function or performance measure to be maximized, and a model of the external environment. Backpropagation is used to calculate a "derivative of utility" summed across all future times with respect to current actions. These derivatives are then used to adapt the ANN which outputs the actions, or to adapt a schedule of actions [185, 186]. But this approach requires a model of the external environment which must be noise-

free and exact. Also, it requires calculations backwards through time, which is not consistent with true real-time learning [110, Chap.2].

In adaptive critic methods, the user supplies a function or measure to be maximized. The long term optimization problem is solved by adapting an additional ANN, called a critic network, which evaluates the progress that the system is making. The network which outputs the actions is adapted to maximize this secondary utility function in the immediate future [110, 187].

Although successful, the ANN suffers from such weaknesses as its parameters generally lacking explicit physical meaning, and it can not effectively utilize linguistic information about the behavior of the systems of interest that may be provided by field experts.

The weaknesses of the ANN are exactly the strong points of another approach in the class of intelligent control techniques — the fuzzy logic approach.

2.2.2 Fuzzy Sets and Fuzzy Logic

The idea of the fuzzy set was introduced by L.A. Zadeh in 1965 [204] as a generalization of the idea of an ordinary or crisp set. Crisp sets allow only full membership or no membership at all, while fuzzy sets allow partial membership, i.e., fuzzy sets allow for the description of concepts in which the boundary between having or not having a certain property is not sharp. The application of the fuzzy set concept in traditional logic resulted in the development of fuzzy logic, which was first outlined in the seminal paper [206] by Zadeh in 1973. Since then, the theory of fuzzy sets and fuzzy logic (or fuzzy logic in short) has found many applications in a variety of subject fields, such as in control engineering, artificial intelligence, management/society, and so on [101, 130, 132, 156, 163, 187, 197, 198].

As pointed out by Pedrycz [130]: *The role of fuzzy sets as a general and fundamental principle of processing of nonnumerical (nonpointwise) information is not*

restricted to a specific field and, essentially, can be embedded, to a certain extent, into almost any area requiring human expertise or calling for the formalization of linguistic specifications and that of relevant knowledge. Terano et al. [163] presented an excellent overview of applications of the theory of fuzzy sets and fuzzy logic in various fields, which the interested reader may consult for more details.

In this work, our primary interest lies in the regime of control engineering and we restrict out scope of discussion accordingly.

For most engineering systems, there are two important information sources: sensors which provide numerical measurements of variables, and human experts who provide linguistic instructions and descriptions of the systems. The information of the first type is called numerical information, and the second is called linguistic information, which is usually represented in fuzzy terms. Conventional engineering control approaches and ANNs, can make use of only the numerical information and have difficulty incorporating linguistic information. In contrast to this, the fuzzy logic approach provides a systematic and efficient framework for incorporating linguistic descriptions of human expert knowledge into the design of automatic controllers.

The literature on fuzzy logic based identification and control has been growing rapidly, making it unfeasible to present a comprehensive review of this vast field. Here, rather than a comprehensive review, we synthesize the published works and give a brief introduction to the history and applications of fuzzy logic based identification and control. More detailed surveys and commentaries may be found in [60, 67, 88, 97, 102, 109, 164, 166].

Generally, if a fuzzy logic system is used to model an unknown plant, it is an identifier in the conventional sense, and if it is used to generate control signals for the plant, it assumes the role of a controller.

Fuzzy logic control systems were proposed in the early seventies [27, 205, 206].

Credit is due mainly to the seminal article by Zadeh in 1973 [206] on a new approach to the analysis of complex systems that the basic theory upon which fuzzy logic controllers would be built was formulated. The first fuzzy logic controller (FLC) was developed and published by Mamdani and Assilian the following year [99, 103, 104], to control a laboratory model steam engine. Since then, FLC has been successfully applied to control various complex and ill defined systems and industrial processes, such as aircraft flight control [86], automobile speed control [113], train operation system control [201], continuous casting plants control [10], polymer extrusion production control [45], cement kiln control [169], water purification process control [199], and many others [11, 87, 104, 153, 156, 210]. The successful application of FLC to a variety of consumer products [129, Chap.4] such as: FLC tuned televisions, focused and stabilized camcorders, washing machines, defrosting refrigerators, and scheduled traffic lights, has impressed the general public and contributed to the current interest in fuzzy logic theory.

The impressive successes of the fuzzy logic control approach is based on intrinsic advantages of this approach over conventional control approaches for many applications. As indicated by Mamdani, et al,

- *The main merit of a FLC is that it gives the most efficient knowledge representation method that can be devised for rule based systems that deal with continuous variables* [102].

- *The basic idea behind this approach was to incorporate the "experience" of a human process operator in the design of the controller. From a set of linguistic rules which describe the operator's control strategy a control algorithm is constructed where the words are defined as fuzzy sets. The main advantages of this approach seem to be the possibility of implementing "rule of thumb" experience, intuition, heuristics, and the fact that it does not need a model of*

the process [73].

For example, human operators can very successfully control complex, highly uncertain systems, such as aircraft, without reference to a precise (or any) mathematical model. If asked what control strategies they followed, they describe their expertise linguistically, in a fashion which can be summarized into a set of fuzzy control rules in linguistic terms, e.g.,

IF SPEED IS TOO SLOW, THEN INCREASE FUEL, ...

Although these linguistic descriptions and fuzzy control rules are not precise and may even not be sufficient for constructing a successful automatic controller, they certainly provide very important information about the system behavior and the ways to control the system, which can be effectively integrated into controller designs by the fuzzy logic approach.

From another point of view, in many situations if precise solutions are sought the problems may become intractable. A typical example of this type is the problem of parking a car [207]. We solve this problem every day without making any measurements. We can do this because the final position of the car is not specified with precision. If it were, the cost of solution would be prohibitive. As pointed out recently by Zadeh [207] :

- *In effect, it is the human ability to exploit the tolerance for imprecision that makes it possible to achieve tractability, robustness, and low solution cost. This is what the conventional methods of system analysis and control fail to do.*
  This is exactly what the fuzzy logic approach is all about.

From the aspect of control engineering, linguistic information from human experts can be classified into two classes [130, Chap.8],[184, Chap.8] :

1. Describe control actions from given system states. For example, consider the situation of driving a car in which the linguistic information of this kind is in

the form of "IF the speed of the car is slow, THEN give more gas".

2. Describe the behavior of the unknown plant. With the car again, the linguistic information of this second kind is given as "IF more gas is given, THEN the speed of the car will increase".

Classically, fuzzy logic controllers have been designed by directly utilizing linguistic information of the first type, and an overwhelming body of literature deals with a form of nonlinear fuzzy proportional-derivative-integral control for set point regulation problems, such as the applications presented, among others, in [89, 98, 99, 101, 103, 156, 164, 196, 209]. But this kind of fuzzy logic control approach generally lacks formal synthesis techniques which guarantee the basic requirements of global stability and acceptable performance. The design has been ad hoc, and controller parameters often must be manually adjusted by trial and error. These weaknesses have drawn some criticism of the fuzzy logic approach as not being a rigorous science [126, 127].

To address these difficulties, efforts have been made to develop stability analysis techniques for fuzzy logic systems, to integrate training algorithms into fuzzy logic systems to automatically adjust their parameters (and structures) based on numerical information, and to integrate conventional synthesis and design techniques into the design and analysis of fuzzy logic control systems.

To study the stability of fuzzy logic systems, various approaches have been developed, which include, among others, the following examples.

- Kickert and Mamdani (1978) [73] developed a multilevel relay model of their proposed fuzzy controller and used classical describing function techniques to study the stability problem. But the describing function method is generally used to determine the existence of periodic oscillations in nonlinear systems. It is an approximate method at best and does not directly address the issue

of stability.

- Braae and Rutherford (1979) [22] developed the notion of the so called granularized state-space, and proposed a method of analysis based on systematic mapping of the process state, using control rules as transition functions.

- Kiszka et al. (1985) [74] proposed an energistic approach which was based on a measure of the energy of a closed loop fuzzy dynamic system. In this approach, the notions of stability and robustness are intertwined, and the closed loop system is considered robustly stable if a fuzzy energy function can be found consistently decreasing along the solution trajectories of the system.

- Chen (1989) [31] applied the concept of cell-to-cell mapping to the analysis of stability of fuzzy control systems. See also Kang (1993) [68]. This approach, however, requires an accurate mathematical model of the process, which contradicts the basic premise of the fuzzy logic control approach, i.e., to deal with a complex and/or ill-defined system without knowing a priori its precise mathematical model. Also, well defined systems may be dealt with more efficiently with the many powerful and well developed conventional identification and control approaches.

- Langari and Tomizuka (1990) [85] presented a stability analysis approach for fuzzy control systems by applying Lyapunov's direct method. A technique of this kind was also seen in [161] by Tanaka and Sugeno (1992).

- Wang (1993) [182] proposed an adaptive fuzzy logic system design scheme following the Lyapunov synthesis approach, and the resulting closed loop system was stable in Lyapunov sense. Similar adaptive FLS design procedures have also been reported in [154] by Su and Stepanenko (1994), and [152] by Spooner and Passino (1996).

A fuzzy logic system equipped with a training algorithm is called a self-organizing fuzzy logic system by Pedrycz [129], Procyk and Mamdani [134], Langari and Berenji [187, Chap.4] et al., and called an adaptive fuzzy system by Wang [184]. This kind of fuzzy logic system is constructed from a set of linguistic IF-THEN rules using fuzzy logic principles, and the training algorithm adjusts the parameters (and structures) based on numerical information. As outlined by Wang (1994) [184, Chap.1], there are two strategies for combining numerical and linguistic information using adaptive fuzzy systems:

- *Use linguistic information to construct an initial fuzzy logic system, and then adjust the parameters of the initial fuzzy logic system based on numerical information. The final fuzzy logic system is, therefore, constructed based on both numerical and linguistic information.*

- *Use numerical information and linguistic information to construct two separate fuzzy logic systems, and then average them to obtain the final fuzzy logic system.*

In this work, we are only concerned with the first strategy.

In the literature, various techniques for adjusting fuzzy logic system parameters have been proposed. The popular approaches include gradient based approaches similar to the backpropagation algorithm for artificial neural networks [61, 62, 178], least squares estimation techniques [64, 159, 181], Lyapunov synthesis approaches [154, 182, 184], and a fuzzy version of the signal Hebbian learning law [130, 173]. Other training techniques include, among others, nonlinear programming techniques [159], an orthogonal least squares technique [179], nearest neighborhood clustering [184, Chap. 6], a table–loop up scheme [180], heuristic stochastic optimization algorithm [15], and convex programming techniques for linear matrix inequality problems [176].

Research on integrating conventional design techniques into direct fuzzy logic control system design utilizing the first type of linguistic information, include, among many others, the following examples:

- Palm (1992) [123], Kung and Liao (1994) [84] combined fuzzy logic control systems with sliding mode control techniques to form fuzzy-sliding mode controllers for tracking control of nonlinear systems.

- Vachtsevanos et al. (1992, 1993) [171, 172] presented a fuzzy logic controller design approach which was based on a partitioning of the state space into small rectangles called cell-groups, and quantization of the states and the available controls into finite levels or bins. Membership functions were then assigned for the state and controls. The transition from one conditional subspace to another was accomplished via a center-point mapping of the cell groups, under the applied action of each IF-THEN rule. But there was no stability analysis presented in their works.

- Pedrycz (1993, 1995) [129, 130] proposed a controller structure called the "hybrid fuzzy controller" which blended fuzzy and PID controllers. The main idea was that the fuzzy controller was superb for a far away control where its nonlinear characteristics could generate a fast dynamic response. When approaching to the setpoint, the role of the fuzzy controller diminished and the control action was taken over by the PID element. The switching between far away and nearby control was performed by an additional fuzzy controller making a relevant selection between the corresponding control action of the fuzzy and PID controller.

- Wang (1993) [182] presented a stable direct adaptive fuzzy controller by following a Lyapunov synthesis approach, for tracking control of a general high order nonlinear continuous system.

- Birdwell and Wang (1994) [15] presented a so-called fuzzy-PID controller in which a fuzzy logic system was used to tune the gains of a conventional PID controller where the fuzzy rule base was tuned using a heuristic stochastic optimization algorithm. Sufficient conditions for system stability were given.

- Qin and Borders (1994) [137] proposed a multiregion fuzzy logic controller for nonlinear process control, where the process to be controlled was divided into fuzzy regions such as *high-gain, low-gain, ...*, based on prior knowledge. A fuzzy controller was then designed based on this regional information, and during operation an auxiliary process variable was used to detect the process operating regions.

- Hwang and Tomizuka (1994) [53] presented a fuzzy variable structure control system. For this type of control system, which implements different control laws in different regions of the state space divided by a set of boundary manifolds, the control input switches from one control law to another when the state crosses the boundary manifolds. Hwang and Tomizuka proposed fuzzy rule based algorithms for smoothing the control input when switching at the boundary manifolds to avoid exciting high frequency dynamics.

The second type of linguistic information is analogous to conventional identification problems, and is called *fuzzy modeling* or *fuzzy identification* [129, 159, 165].

Most fuzzy modeling problems seen in the literature are concerned with representing or approximating relationships between input and output variables in the form of some functional dependencies [131], and are more or less based on the seminal works by Zadeh in 1973 [206] and Takagi et al. in 1985 [159]. A notable exception comes in the form of fuzzy relational models proposed by Pedrycz (1984, 1993) [128, 129], which is more general than the concept of function and is to *characterize links (associations) between some objects distributed in several different spaces*

or defined in a Cartesian product of the same space [130, App. B]. Recently, a new modeling framework termed fuzzy multimodels was presented in [131] by Pedrycz (1996), which produces an environment assuring successful interaction (when function oriented or relation focused models are not sufficient) between several relational or functional constructs and allows for their efficient utilization.

In this work, we restrict our scope of fuzzy modeling, or fuzzy identification, to functional models, and leave the study of fuzzy relational models and fuzzy multimodels, as well as development of correspondent control strategies to future consideration.

When linguistic information of the second type is used in control applications, it is sometimes called fuzzy model based control [64], where the plant is modeled or identified with a fuzzy logic system and a controller is constructed based on the fuzzy model, often with the help of conventional controller synthesis and design techniques. There have been several approaches reported to fuzzy model based control.

- A strategy where at each discrete time step a linear controller was designed on the basis of a linearization of the fuzzy model at the current operating point, was used in [94, (1992)], [157, (1986)], [158, (1992)]. This gave a nonlinear controller which is conceptually simple, but suffered from drawbacks similar to those of conventional gain scheduling approaches outlined in section 2.1.

- Wang and Vachtsevanos (1992) [173] presented a methodology for the design of a learning fuzzy logic control system which was based on the concept of indirect control approach where selection of control parameters relied on the estimates of process parameters. The control law consisted of an on-line fuzzy identifier, desired transition model and fuzzy controller. The fuzzy version of the signal Hebbian learning law was introduced for identifying the unknown

plant. There was no system stability analysis associated with this approach.

- Jang (1993) [63] proposed an adaptive-network-based fuzzy inference system, which was a fuzzy inference system implemented in the framework of adaptive networks. By using a hybrid learning procedure, the proposed fuzzy system could construct an input-output mapping based on both human knowledge and stipulated input-output data pairs, and could be used to identify nonlinear components on-line in a control system. Again, there was no system stability analysis presented in this work.

- Johansen (1994) [64] presented a nonlinear controller design and analysis approach which was based on a fuzzy model of MIMO dynamical systems, and the closed loop system was claimed to be globally stable and robust with respect to unstructured uncertainty such as modeling error and disturbances. But this approach was not applicable to systems with strong non-minimum phase features, or in which the unstructured uncertainty was large.

- An indirect adaptive tracking control strategy was proposed by Wang (1994) [184, Chap.8] and Su et al. (1994) [154] separately, where fuzzy logic systems were used to model the unknown plants. In both works, the global stabilities of the closed loop systems were established in the Lyapunov sense.

- Wang et al. [174, 175, (1995)], [176, (1996)] presented a design methodology for stabilization of a class of nonlinear systems, where a nonlinear plant was modeled by a fuzzy logic system and then a model based fuzzy controller design utilizing the so-called "parallel distributed compensation" technique was employed. The main idea of the controller design was to derive each control rule so as to account for each rule of a fuzzy system. The stability analysis and control design problems were reduced to linear matrix inequality (LMI) problems, which could be solved by convex programming techniques.

Apart from the above applications where fuzzy logic systems have been used basically for low level, set-point-oriented problems, they also can be applied to high-level, task-oriented control functions. A simple yet clear example of this kind is a method called *supervisory control* proposed by Pedrycz (1993,1995) [129, 130], where the function of a fuzzy controller was to switch between several *local* controllers. Each of these local controllers was tuned to perform well only in a limited range of its input variables. An industrial application of this control methodology at "General Electric" was also reported in 1995 by Bonissone et al. [17], where fuzzy logic systems were used in high-level, supervisory roles to complement low-level conventional controllers.

2.2.3 Remarks

1. Fuzzy Logic System vs. Artificial Neural Network

From the previous discussion, it is seen that similarities exist between fuzzy logic systems and artificial neural networks. They share a common objective, which is to emulate the operation of the human brain. *In some sense, artificial neural networks try to emulate the "hardware" of the human brain, whereas adaptive fuzzy systems try to emulate the "software" in the human brain* [184, Chap.7]. They can both handle extreme nonlinearities in the system. Both techniques allow *interpolative reasoning*, which frees us from the true/false restriction of logical systems such as those used in symbolic artificial intelligence [187, Chap.4]. For example, once an ANN has been trained for a set of data, it can interpolate and produce answers for cases not present in the training data set. Similar properties hold for a FLS. Also, the weighted average scheme of FLSs and the sum of products of the ANNs are similar in principle.

However, the two methodologies have a fundamental difference, which is that

the fuzzy logic system takes linguistic information explicitly into consideration and makes use of it in a systematic way, whereas the artificial neural network does not. Also, adaptive fuzzy systems are said to constitute a much larger functional space than artificial neural networks [184, Chap.7]. More details of comparison between FLS and ANN may be found in [184, 187].

In recent years, there has been a growing interest in integrating FLS with ANN, such as the examples presented, among many others, in [28, 129, 130, 152, 187, 195]. This fusion allows use of a humanly comprehensible expression of the knowledge used in control in terms of fuzzy control rules; the fuzzy controller learns to adjust its performance automatically using an ANN structure [187, Chap.4].

2. Fuzzy Logic System vs. Evolutionary Computation

During the last two decades, there has been a growing interest in algorithms which are based on the principle of evolution (survival of the fittest), which are generally referred to as "evolutionary computation". The best known algorithms in this class include genetic algorithms, evolutionary programming, evolution strategies, and genetic programming [132, Chap.1.1]. Since the best known evolutionary computation techniques are genetic algorithms, very often the term *evolutionary computation* and *genetic algorithm* are used interchangeably in the literature.

As pointed out by Davis and Steenstrup [35, Chap.1] (see also [132, Chap.1]), *... the metaphor underlying genetic algorithms is that of natural evolution. In evolution, the problem each species faces is one of searching for beneficial adaptations to a complicated and changing environment. The "knowledge" that each species has gained is embodied in the makeup of the chromosomes of its members.*

Genetic computations are aimed at finding a global maximum of a function of many variables through performing a genetic-inclined search of the space [130, Chap.3]. This methodology has been integrated with FLS and ANN to increase

computational efficiency [130. 132, 194], where [132] (Pedrycz ed., 1997) presented a comprehensive and updated collection of both fundamental tutorials and cutting edge research of the fusion of fuzzy logic system and evolutionary computation. which interested readers may consult for more details.

2.3 Modeling and Control of Flexible Link Manipulators

Because of their potential use in space applications as well as in new generations of industrial robots. there has been increasing interest in designing light weight robotic systems. which usually possess flexible links. Flexural behavior in an already highly nonlinear robotic system considerably complicates analysis and design. and demands elaborate control schemes with attendant requirements for sophisticated hardware/software.

In this work. flexible link manipulators are considered as examples of challenging and important practical systems on which our fuzzy logic based identification and control methodologies can demonstrate their merits.

A flexible link manipulator was constructed to serve as the experimental testbed for the theoretical developments. In addition, the dynamic model of a flexible link manipulator is developed and used as, among others, an important numerical example for simulation studies of our theoretical developments. Specifically, the end point position control of the robotic manipulator is of interest in this research.

As modeling and control of flexible link manipulators are not the core subjects of this research. only a very brief review will be presented in the following.

2.3.1 Introduction

Today's commercial robots share the common features of being heavy and bulky. They operate at low speed, have low load to weight ratio and high energy consumption, which lead to low productivity and low efficiency, and therefore, are inapplicable for situations requiring light weight structures, low energy consumption, high maneuverability, easy transportability, and smaller, lighter actuators, such as in space applications. To replace the heavy structure with light weight structure is generally desirable to advance the state of the art of robotics. The use of light weight structures will certainly lead to the use of flexible links. This is one of the reasons for the initiation in the seventies [18, 19] of work in this area and there has been increasing interest over the past decade in the analysis and control of flexible link robotic manipulators [16, 20, 26, 34, 47, 48, 82, 111, 117, 136, 142, 143, 168, 170, 193, 200, 203, 208].

Some of the undesirable features of current robots originate from their control schemes. For example, to position the end point of a manipulator, the desired location is transformed into equivalent angles that each of the robot joints must assume with real time kinematic computation, and every joint is then driven to the specified angle simultaneously by employing a PI servo loop which is made up of a torquer and a collocated sensor, e.g., an encoder. The accuracy of the end point is based on the rigidity of the robot links which must be guaranteed with stiff and therefore bulk structures.

Compared with their rigid link counterparts, flexible link manipulators are much lighter and have many desirable properties, such as low energy consumption, high maneuverability, easier transportability, and smaller, lighter actuators. But these desired features are somewhat vitiated by the much more stringent requirements on their control system design imposed by the flexibility. The end point accuracy of a flexible link is severely degraded due to structural deformation and movement induced vibrations. To achieve maximum accuracy in positioning a manipulator's

working end (i.e., the tip), sensors often must be used to measure the tip position, while the control torque that is based on the measurements of the tip position is applied at the other end of the link. This scheme results in noncollocated sensors and actuators, which greatly complicates the controller design compared to the situation of collocated sensors and actuators where stable servo control is guaranteed by the collocation [26]. In addition, the mathematical models of a flexible link robotic system can be highly nonlinear. This has resulted in increased research activity into methods of actively influencing flexible manipulator behavior, as mentioned above.

A practical manipulator may be multidimensional with multiple links. But such a robotic system which is additionally elastic is too complex to be dealt with thus far. The basic analysis and control issues can be exposed and should be solved in the context of a one dimensional case, i.e., a single link planar manipulator case. It is observed that, currently, most published works are basically at this stage. In this work, we restrict our scope to the single link planar manipulator. This system is indeed sufficiently complex, with unknown and nonlinear elements, to pose a real challenge to the most advanced control strategies, as we shall see.

2.3.2 Dynamics of Flexible Link Manipulator

A flexible link is a distributed parameter system, and rigorous representation of such a system usually requires a set of partial differential equations, which can be derived using one of many available approaches (described in standard dynamics text books), such as the Hamilton Principle and Lagrangian Dynamics.

The derived partial differential equations (PDEs) must be solved. Generally speaking, there are only a few simple and special forms of PDEs which can be solved explicitly, i.e., to obtain their closed form solutions [122, 141]. In most practical situations, this is not possible, and the PDEs are discretized and solved in approximate form. Tzafestas presented a good survey on various discretization methods [167,

Chapter 1], such as modal expansion methods, the Green's function method [2], the finite element method, and so on.

Modal Expansion Method

The modal expansion method has been widely used in solving dynamic problems. The solution is expanded with system eigenvalues and eigenvectors, and results in a series of infinite order. Thus the PDEs can be transformed into set of ordinary differential equations (ODEs) of infinite order. But due to limitations of computer power, sensor inaccuracy and system noise (not to mention physical reality!), the solution is approximated by truncating the infinite series and retaining only some of the terms, that is, the solution is approximated with a series of finite order. Thus, the PDEs can be transformed into a set of ODEs of finite order. Further, the ODEs may be rearranged into state space expression or transfer function, which are convenient for control purposes.

The above procedure is one of the most popular approaches to the modeling of flexible robot links [16, 18, 19, 20, 26, 34, 47, 48, 82, 111, 136]. In most cases, a flexible link is modeled as a clamped-free or pinned-free Euler-Bernoulli beam superposed with a rigid body rotation. Hastings and Book indicated [48] that the clamped-free Euler-Bernoulli beam model agreed better with measured responses than that of a pinned-free beam model. Also, they believed that the first two vibration modes dominated the system responses, a view which was also shared by Zaki [208].

To avoid nonlinearities which cause immense difficulties in applying conventional controller design techniques, radical simplifications have commonly been made in published works, which generally include the assumptions of small deflections of the

[2]Green function approach gives closed form solutions of PDE's. But in general situation, Green function itself may be expanded with system eigenvalues and eigenfunctions [90].

flexible beams, small angular velocities, ignoring rotary inertia of the beam cross-section and end-point payload, ignoring shear deformation effects, and nonlinear friction.

Finite Element Method

Another approach to the discretization of distributed parameter systems is the finite element method (FEM) [211]. This method has been primarily used to solve problems of large scale and complex systems, but it has seen occasional use in application to the modeling of flexible links, e.g., [13, 170], in which each flexible link is assumed to be composed of beam type finite elements attached to each other with the first and the last satisfying corresponding system boundary conditions. The deformed shapes of the link are described by using a set of perturbation coordinates for the entire link and a set of shape functions that characterize the displacements of link points located between the finite element nodes. The main drawback of a FEM model is its inherent computation burden.

Lumped Parameter Method

In contrast to the distributed parameter system which is assumed to occupy a certain spatial domain, the lumped parameter system is assumed to be concentrated at a spatial point. For this kind of system, ordinary differential equations may be directly obtained. An example of this approach is in [117], in which Nelson, et al. modeled a flexible link as a lumped mass-spring-mass system and obtained the ODEs of the system directly. This model ignored some significant features of real flexible robot arms. For example, the distributed flexibility and the delay effects associated with wave propagation along the length of the beam are neglected. This may be one reason that the lumped parameter system has not been widely applied in modeling flexible links.

There are more approaches for dynamic modeling of flexible link manipulators, of which mention is not made here. Interested readers may consult an excellent review on this subject in [2]. For the purpose of the proposed research, the modal expansion of an Euler-Bernoulli beam is deemed to be an adequate first approach to serve as a computer simulation tool.

2.3.3 Position Control of Flexible Link Manipulators

The objective of position control is to command the end-effector of the robotic manipulator to follow a desired trajectory, or simply to maintain a desired position. Various control schemes have been developed for this purpose.

Most published works linearized dynamic models of the manipulators, and applied linear control techniques.

For nonvarying and known payloads, Book and Majette (1983) [20] used a pole placement (linear state feedback) method to design a linear controller. Cannon and Schmitz (1984) [26] introduced an experimental approach to identify the coefficients of the system's non-minimum phase transfer function based on frequency response analysis, and further developed a linear quadratic Gaussian controller. They also observed that if the system mathematical model did not match the system's actual behavior well, the control system could be unstable. Other examples of this kind may be found in [16, 47, 82].

To compensate for unknown/varying payload, adaptive control techniques have been used to estimate the payload, such as in [117, 142, 143, 200, 203]. Rovner et al. (1987,1988) [142, 143] proposed load adaptive control algorithms based on the self-tuning technique, where dynamic models of the flexible links were identified with a recursive least squares algorithm, and controllers were obtained based on linear quadratic Gaussian synthesis procedures. Yurkovich et al. (1989) [203] proposed a similar adaptive control scheme, where, in contrast to others, they incorporated

end-point acceleration (instead of position) into control laws.

In order to use conventional inverse tracking control techniques, Davis (1988) [34] proposed a manipulator mechanism characterized by a collocated position sensor and force actuator, which resulted in a minimum phase dynamic model. Similar efforts have also been made in [111, 150].

Robust control techniques have also been used. For example, Qian and Ma (1992) [136] proposed a variable structure sliding mode control scheme for end-point tracking control. Franke (1986) [38] proposed a robust variable structure controller in the presence of plant uncertainties for a distributed parameter system.

Usually the controller of a flexible structure is designed based on its reduced order model with the potential for excitation of the residual modes by control signals, termed *spillover*, which can cause unwanted, noise-like vibration in the system. To avoid this problem, Lin (1992) [92] proposed a robust controller based on addition of a parallel or cascade connected residual mode filter with an observer.

In recent years, the ANN has been used in position control of flexible link manipulators. For example, Newton and Xu (1993) [118] introduced an ANN based on on-line learning control for a flexible space manipulator. A similar scheme was also presented in [160]. Donne and Özgüner (1994) [37] proposed a control scheme for flexible manipulator with partially known dynamics, where an ANN was used to identify the unknown flexible dynamics. See also [93] for a similar control strategy.

The fuzzy logic approach has been found to be effective in control of flexible link manipulators. For example, Tzes and Kyriakides (1993) [168] proposed a control scheme for flexible link manipulators, where the region of the eigenvalue space was partitioned into fuzzy cells and membership functions were assigned to the fuzzy sets of the eigenvalue universe of discourse which was utilized by a fuzzy controller to infer control actions. Akbarzadeh-Totonchi (1994) [1] proposed a knowledge based, model free fuzzy control scheme with a two level hierarchical structure, where the

higher level monitored link behavior and extracted features in fuzzy terms, while the lower level used this information to control the system. Lee et al. (1994) [91] also proposed a knowledge based, model free fuzzy controller for end-point position control.

Alasty (1997) [2] presented a detailed review of modeling and control of flexible link manipulators, which interested readers may consult for more details.

Chapter 3

Brief Review of Fuzzy Sets and Fuzzy Logic

3.1 Introduction

The idea of the fuzzy set was introduced by L.A. Zadeh in 1965 [204], to allow imprecise and qualitative information to be expressed and used in an exact way, and is, as the name implies, a generalization of the ordinary notion of a set. An example is appropriate to illustrate this [164]: Suppose it is required to specify linguistic measures of temperature on the closed interval [100°C, 200°C], and that such a measure is $\mathcal{T} \triangleq$ {temperatures about 150°C}. An ordinary set which defines this can be expressed in terms of a membership function, $\mu_{\mathcal{T}}$(T), which can take values of either 0 or 1. If a temperature T is not a member of the set $\mathcal{T}$, i.e., T$\notin \mathcal{T}$, then $\mu_{\mathcal{T}}$(T)=0; if it is a member of $\mathcal{T}$, i.e., T$\in \mathcal{T}$, then $\mu_{\mathcal{T}}$(T)=1. Graphically, $\mu_{\mathcal{T}}$ may be defined as the rectangular function shown in fig.3.1(a). A fuzzy set which expresses the corresponding idea has a membership function which has a continuum of grades of membership, i.e., it takes all values in [0, 1], and might thus be as shown in fig.3.1(b). The ordinary set has an abrupt transition from membership to non-membership, i.e., an item is either a member or not a member of a set.

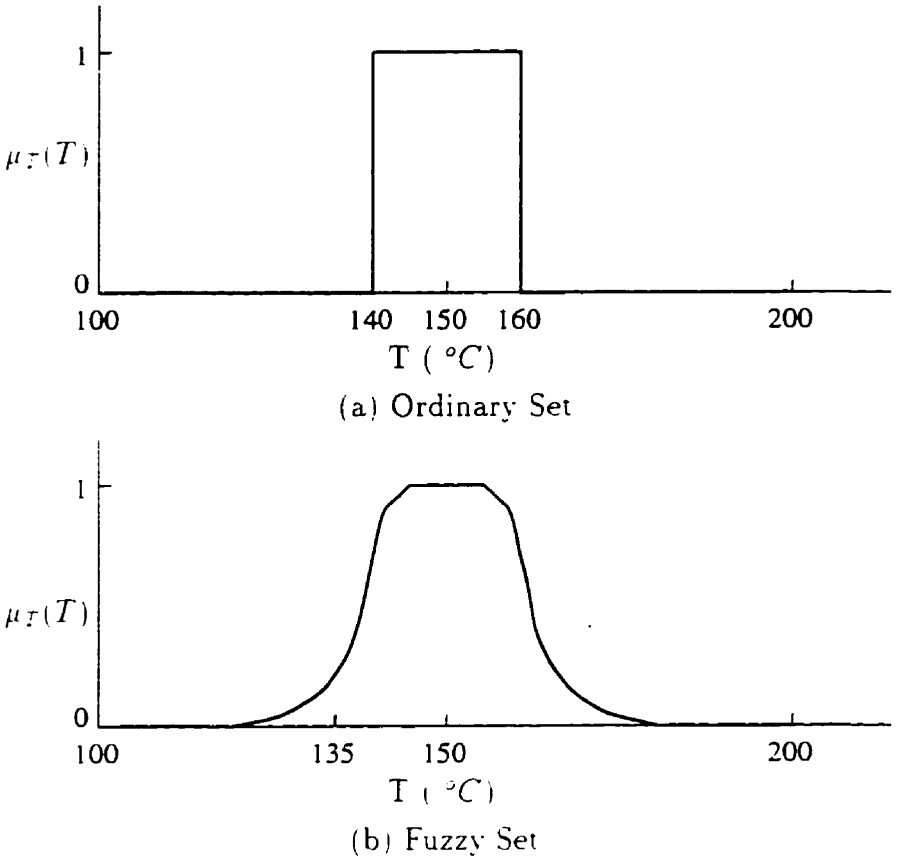


Figure 3.1: The Concept of Fuzzy Set

The fuzzy set, on the other hand, allows the qualitativeness of the measure to be reflected in a gradual membership transition, i.e., an item is allowed to belong to a set to a certain degree. For example, consider T= 135°C. In the case of the ordinary set, this temperature is not a member of the set $\mathcal{T}$, i.e., $\mu_{\mathcal{T}}(135°C) = 0$; in the case of the fuzzy set, it is a member of $\mathcal{T}$ with grade of membership 0.25, i.e., $\mu_{\mathcal{T}}(135°C) = 0.25$. Using this idea, qualitative information can be represented mathematically and handled in a systematic and rigorous manner.

The application of the fuzzy set concept to traditional logic resulted in the development of fuzzy logic, which was first outlined in the seminal paper [206] by Zadeh in 1973. Since then, the theory of fuzzy sets and fuzzy logic (or fuzzy logic in short) has found many applications in a variety of subject fields, such as in control engi-

neering, artificial intelligence, management/society, and so on [163]. *In fact, any field X can be fuzzified and labeled fuzzy X by replacing the concept of a set in X by the concept of a fuzzy set* [184, in the Foreword by Zadeh].

Because the primary goal of this development is in the regime of control engineering, we are interested only in the theory of fuzzy sets and fuzzy logic as they pertain to this area, and which is introduced in the following.

3.2 Basics of Fuzzy Sets and Fuzzy Logic

In this section, we present only a very brief introduction to those aspects of the theory of fuzzy sets and fuzzy logic which are relevant to the development of this work. However, we will indicate references in which more comprehensive and in-depth information is to be found. Following now is a collection of basic concepts summarized from [88, 129, 184, 204, 206, 212].

3.2.1 Fuzzy Sets

Universe of Discourse: Let $\mathcal{U}$ be a collection of objects denoted generically by $\{u\}$, which can be discrete or continuous. $\mathcal{U}$ is called the *universe of discourse* and u represents a generic element of $\mathcal{U}$.

Fuzzy Set: With $\mathcal{U}$ a universe of discourse, a fuzzy set U in $\mathcal{U}$ is a set of ordered pairs

$$U \triangleq \{(u, \mu_U(u)) \mid u \in \mathcal{U}\} \ . \tag{3.1}$$

in which μ_U is called the membership function of U and takes values in the interval $[0, 1]$, and $\mu_U(u)$ represents the grade of membership of u in U. Let U_1 and U_2 be two fuzzy sets in $\mathcal{U}$ with membership functions μ_{U_1} and μ_{U_2}, respectively. These are *equal*, written as $U_1 = U_2$, if and only if $\mu_{U_1}(u) = \mu_{U_2}(u)$ for all $u \in \mathcal{U}$.

Support: The support of a fuzzy set U is the crisp, i.e., non-fuzzy, set of all points

u in $\mathcal{U}$ such that $\mu_U(u) > 0$.

Fuzzy Singleton: A fuzzy set whose support is a single point in $\mathcal{U}$ is referred to as a fuzzy singleton.

The Extension Principle: Let $\mathcal{U}$ and $\mathcal{V}$ be two universes of discourse and f a mapping from $\mathcal{U}$ to $\mathcal{V}$. For a fuzzy set U in $\mathcal{U}$, the extension principle allows us to define a fuzzy set V in $\mathcal{V}$ by

$$V \triangleq \{(v, \mu_V(v)) \mid v = f(u), u \in \mathcal{U}\} , \tag{3.2}$$

where

$$\mu_V(v) \triangleq \begin{cases} \sup_{u \in f^{-1}(v)}[\mu_U(u)], & if\ f^{-1}(v) \neq 0; \\ 0, & otherwise. \end{cases} \tag{3.3}$$

where f^{-1} is the inverse of f.

The extension principle is one of the most basic concepts of fuzzy set theory, and is used to generalize crisp mathematical concepts to fuzzy sets.

3.2.2 Operations on Fuzzy Sets

Union: The union of two fuzzy sets U_1 and U_2 is a fuzzy set U_3, written as $U_3 \triangleq U_1 \cup U_2$, whose membership function is pointwise defined for all $u \in \mathcal{U}$ by

$$\mu_{U_3}(u) = \mu_{U_1 \cup U_2}(u) \triangleq \max\{\mu_{U_1}(u), \mu_{U_2}(u)\}. \tag{3.4}$$

Intersection: The intersection of two fuzzy sets U_1 and U_2 is a fuzzy set U_3, written as $U_3 \triangleq U_1 \cap U_2$, whose membership function is pointwise defined for all $u \in \mathcal{U}$ by

$$\mu_{U_3}(u) = \mu_{U_1 \cap U_2}(u) \triangleq \min\{\mu_{U_1}(u), \mu_{U_2}(u)\} \tag{3.5}$$

Complement: The complement of fuzzy set U is denoted by $\overline{U}$ and is pointwise

defined for all $u \in \mathcal{U}$ by

$$\mu_{\overline{C}}(u) \triangleq 1 - \mu_C(u) . \tag{3.6}$$

t-norm: A t-norm, denoted by "$\star$", is a two place function from $[0, 1]\times[0, 1]$ to $[0, 1]$, which [129, Chap.1]

- is nondecreasing in each argument, i.e., if $x \leq x'$, $y \leq y'$, then $x \star y \leq x' \star y'$;
- is commutative, i.e., $x \star y = y \star x$;
- is associative, i.e., $(x \star y) \star z = x \star (y \star z)$;
- satisfies the boundary conditions, $x \star 0 = 0$, $x \star 1 = x$;

where x, x', y, y', $z \in [0, 1]$. Typical t-norms are defined as follows [212, Chap.3],

$$\begin{cases} x \star y \triangleq \min\{x, y\} . & (minimum) . \\ x \star y \triangleq xy . & (algebraic\ product) . \\ x \star y \triangleq \max\{0,\ x + y - 1\} . & (bounded\ difference) . \\ x \star y \triangleq \begin{cases} \min\{x, y\} . & if\ \max\{x, y\} = 1 , \\ 0 . & otherwise . \end{cases} & (drastic\ product) . \end{cases} \tag{3.7}$$

s-norm: A s-norm, denoted by "$\dot{+}$", is a two place function from $[0, 1]\times[0, 1]$ to $[0, 1]$, which [129, chap.1]

- is nondecreasing in each argument;
- is commutative;
- is associative;
- satisfies the boundary conditions, $x \dot{+} 0 = x$, $x \dot{+} 1 = 1$;

Typical s-norms are defined as follows [212, Chap.3].

$$\begin{cases} x \dot{+} y \triangleq \max\{x, y\} , & (maximum) , \\ x \dot{+} y \triangleq x + y - xy , & (algebraic\ sum) , \\ x \dot{+} y \triangleq \min\{1, x + y\} , & (bounded\ sum) , \\ x \dot{+} y \triangleq \begin{cases} \max\{x, y\} , & if\ \min\{x, y\} = 1 , \\ 1 , & otherwise , \end{cases} & (drastic\ sum) . \end{cases} \tag{3.8}$$

where $x, y \in [0, 1]$.

3.2.3 Fuzzy Relations and Compositions

Fuzzy Relation : A fuzzy relation F is a fuzzy set in the Cartesian product $\mathcal{U} \times \mathcal{V}$, and is characterized by a membership function $\mu_F(u, v)$, $u \in \mathcal{U}$, $v \in \mathcal{V}$, where $\mathcal{U}$ and $\mathcal{V}$ are two universes of discourse. Formally,

$$F \triangleq \{((u, v), \mu_F(u, v)) \mid (u, v) \in \mathcal{U} \times \mathcal{V}\} . \tag{3.9}$$

Generally, an n-ary fuzzy relation F is a fuzzy set in the Cartesian product, $\mathcal{U}_1 \times \cdots \times \mathcal{U}_n$, and is characterized by a membership function, $\mu_F(u_1, \ldots, u_n)$, $u_i \in \mathcal{U}_i$, $i = 1, \ldots, n$.

$$F \triangleq \{((u_1, \ldots, u_n), \mu_F(u_1, \ldots, u_n)) \mid (u_1, \ldots, u_n) \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_n\} . \tag{3.10}$$

Sup-star Composition : If F and S are fuzzy relations in $\mathcal{U} \times \mathcal{V}$ and $\mathcal{V} \times \mathcal{W}$, respectively, the fuzzy relation in $\mathcal{U} \times \mathcal{W}$ is given by the composition of F and S, which is denoted by $F \circ S$ and is characterized by a membership function $\mu_{F \circ S}(u, w)$, i.e.

$$F \circ S \triangleq \{((u, w), \mu_{F \circ S}(u, w)) \mid (u, w) \in \mathcal{U} \times \mathcal{W}\} . \tag{3.11}$$

with

$$\mu_{F \circ S}(u, w) \triangleq \sup_{v \in \mathcal{V}} \{\mu_F(u, v) \star \mu_S(v, w)\} . \tag{3.12}$$

where $u \in \mathcal{U}$, $v \in \mathcal{V}$, $w \in \mathcal{W}$. The symbol "$\star$" represents the t-norm. The most commonly used sup-star compositions are the sup-min and sup-product compositions, which replace the "$\star$" in eq.(3.12) by "min" and algebraic product, respectively, i.e.,

$$\mu_{F \circ S}(u, w) \triangleq \sup_{v \in \mathcal{V}} \{\min\{\mu_F(u, v), \mu_S(v, w)\}\} , \quad (\sup - \min) \tag{3.13}$$

$$\mu_{F \circ S}(u, w) \triangleq \sup_{v \in \mathcal{V}} \{\mu_F(u, v) \cdot \mu_S(v, w)\} , \quad (\sup - \textit{product}) \tag{3.14}$$

A special case of this is with S, instead of being a fuzzy relation, is a fuzzy set in $\mathcal{V}$. In this case, $F \circ S \triangleq \{(u, \mu_{F \circ S}(u)) \mid u \in \mathcal{U}\}$, and in eq.(3.12), the $\mu_{F \circ S}(u, w)$ becomes $\mu_{F \circ S}(u)$, the $\mu_S(v, w)$ becomes $\mu_S(v)$, and the others remain the same.

3.2.4 Linguistic Variables and Approximate Reasoning

Linguistic Variable : A linguistic variable is a generalized real-valued variable whose values can either be real numbers or be linguistic terms. Formally, a linguistic variable is characterized by a quintuple, $(u, T(u), \mathcal{U}, G, M)$, in which u is the name of a variable; $T(u)$ is the term set of u, which includes the set of permissible real values of u in $\mathcal{U}$ and the set of names of *linguistic values* of u with each value being a fuzzy set defined in the universe of discourse $\mathcal{U}$; G is a syntactic rule for generating the names of linguistic values of u; and M is a semantic rule for associating with each linguistic value its meaning [212].

For example, if *speed* is a linguistic variable defined in a universe of discourse $\mathcal{U} \triangleq [0, 100]$, then its term set $T(speed)$ may be defined as

$$T(speed) \triangleq \{10, 55, \ldots\}, \textit{ or } \{slow, medium, fast, \ldots\} , \tag{3.15}$$

where each term in $T(speed)$ is either a real number in $\mathcal{U}$ or a linguistic value (a fuzzy set) in $\mathcal{U}$. The meaning of the linguistic value, *medium*, could be defined as, say,

$$M(medium) \triangleq \{(speed, \mu_{medium}(speed)) \mid speed \in \mathcal{U}\} . \tag{3.16}$$

where

$$\mu_{medium}(speed) \triangleq \exp[-\frac{1}{2}(\frac{speed - 50}{0.45})^2] \ . \tag{3.17}$$

Generalized Modus Ponens (GMP): It has been suggested in [88, 100, 212] that from the perspective of control engineering, the main tool of reasoning for fuzzy logic control systems is the *generalized modus ponens* rule :

$$\begin{aligned} &premise\ 1\ : && u\ is\ U' \\ &premise\ 2\ : && IF\ u\ is\ U\ \ THEN\ \ v\ is\ V \\ &consequence: && v\ is\ V' \end{aligned} \tag{3.18}$$

where u and v are linguistic variables in the universes of discourse $\mathcal{U}$ and $\mathcal{V}$, respectively. U and U' are fuzzy sets in $\mathcal{U}$. V and V' are fuzzy sets in $\mathcal{V}$. If the fuzzy sets are replaced with non-fuzzy, or crisp sets, and $U = U'$, $V = V'$, this becomes the modus ponens rule of traditional logic [212, Chap.9], which is the reason that it is called the *generalized modus ponens*.

An example of the GMP is [212, Chap.9],

$$\begin{aligned} &premise\ 1\ : && This\ tomato\ is\ very\ red \\ &premise\ 2\ : && IF\ a\ tomato\ is\ red\ \ THEN\ \ the\ tomato\ is\ ripe \\ &consequence: && This\ tomato\ is\ very\ ripe \end{aligned}$$

In this example, "*very*" is called a *linguistic hedge* or a *modifier*, a concept which will not be explored here in more detail, as it is not to be used explicitly in this work. Interested readers may refer to [88, 206, 212] for more details.

Fuzzy Implication: Let U and V be fuzzy sets in $\mathcal{U}$ and $\mathcal{V}$, respectively, with $u \in \mathcal{U}$ and $v \in \mathcal{V}$ linguistic variables; a *fuzzy implication*, denoted by $U \rightarrow V$, is a special kind of fuzzy relation in $\mathcal{U} \times \mathcal{V}$ defined as[184]

$$U \rightarrow V \triangleq \{((u,v), \mu_{U \rightarrow V}(u,v)) \mid (u,v) \in \mathcal{U} \times \mathcal{V}\} \ , \tag{3.19}$$

where the membership function $\mu_{U\to V}(u,v)$ conforms to a number of possible interpretations:

$$\mu_{U\to V}(u,v) \triangleq \begin{cases} \mu_U(u) \star \mu_V(v), & (\textit{fuzzy conjunction}): \\ \mu_U(u) \dot{+} \mu_V(v), & (\textit{fuzzy disjunction}): \\ \mu_{\overline{U}}(u) \dot{+} \mu_V(v), & (\textit{material implication}): \\ \mu_{\overline{U}}(u) \dot{+} \mu_{U\star V}(v), & (\textit{proportional calculus}): \\ \sup\{c \in [0,1] \mid \mu_U(u) \star c \leq \mu_V(v)\}, & (\textit{generalized modus ponens}). \end{cases} \tag{3.20}$$

In GMP, eq.(3.18), the fuzzy IF-THEN rule in *premise 2* can be implemented with a *fuzzy implication*, $U \to V$. The different definitions of the membership function in eq.(3.20) correspond to different interpretations or implementations of fuzzy IF-THEN rules that are based on intuitive criteria or generalizations of classical logic [184].

Sup-star Compositional Rule of Inference : The GMP, or sometimes called fuzzy conditional inference [212, Chap.9], is implemented via the *compositional rule of inference*. Formally, for universes of discourse $\mathcal{U}$ and $\mathcal{V}$, let $u \in \mathcal{U}$, $v \in \mathcal{V}$ be linguistic variables, U, V be fuzzy sets in $\mathcal{U}$ and $\mathcal{V}$, respectively, and $U \to V$ be a fuzzy implication in $\mathcal{U} \times \mathcal{V}$. Then for a given fuzzy set U' in $\mathcal{U}$ the sup-star compositional rule of inference asserts that the fuzzy set V' in $\mathcal{V}$ induced by U' is given by the sup-star composition of U' and $U \to V$, i.e.,

$$V' \triangleq U' \circ (U \to V) = \{(v, \mu_{U'\circ(U\to V)}(v)) \mid v \in \mathcal{V}\} . \tag{3.21}$$

where

$$\mu_{U'\circ(U\to V)}(v) \triangleq \sup_{u\in\mathcal{U}} \{\mu_{U'}(u) \star \mu_{U\to V}(u,v)\} . \tag{3.22}$$

This brief review of fuzzy sets and fuzzy set operations is sufficient for the purposes of this work. Further details can be found in the indicated references.

Chapter 4

Fuzzy Logic Systems

4.1 Introduction

In the general literature, the term "fuzzy logic system" is not always precisely defined and frequently refers to almost any system which has something to do with the concepts of fuzzy sets and fuzzy logic. In this work, the scope is restricted to the regime of control engineering, and the meaning and structure of the fuzzy logic system is restricted to this application.

The basic structure of a fuzzy logic system in control engineering is as shown in fig.4.1. Throughout this work, a fuzzy logic system with this structure will be abbreviated as FLS. A FLS is composed of four major components, namely, *a fuzzification interface*, *a fuzzy rule base*, *a fuzzy inference engine* and *a defuzzification interface*. The input and output variables of a FLS are linguistic variables, which assume real values preceding the fuzzification interface and following the defuzzification interface, and take linguistic terms (fuzzy sets) in between, and this FLS can naturally fit into engineering systems. It works as follows: real values of the input linguistic variables are fuzzified into corresponding fuzzy input sets according to certain fuzzification strategies. These fuzzy input sets are then fed into a fuzzy inference engine which, by triggering relevant fuzzy rules in the rule base, maps the

fuzzy input sets to a fuzzy output set which is in turn defuzzified to a real valued output signal.

Since a multi-input, multi-output (MIMO) system can often be decomposed into a group of multi-input, single-output (MISO) systems, only MISO fuzzy logic systems will be considered in this work, as illustrated in fig.4.1.

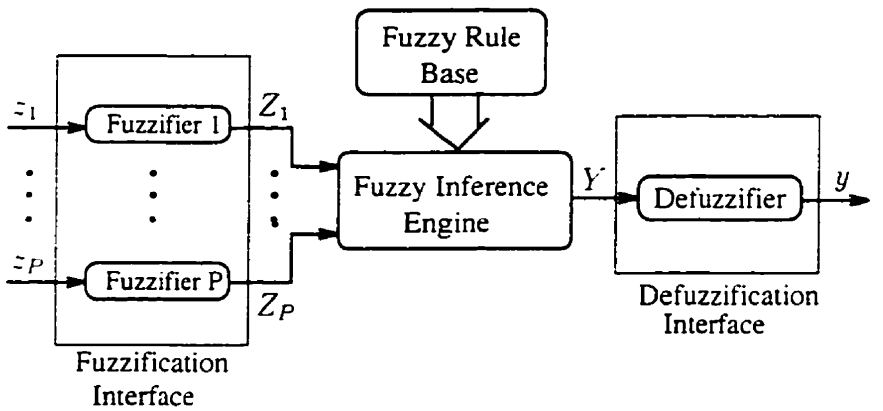


Figure 4.1: Basic Structure of Fuzzy Logic System

The symbols in fig.4.1 are defined as follows: for $p = 1, \ldots, P$, $x_p \in \mathcal{X}_p \subset R$ are linguistic input variables, $y \in \mathcal{Y} \subset R$ is linguistic output variable, $\mathcal{X}_p$ is the universe of discourse of the linguistic variable x_p, $\mathcal{Y}$ is the universe of discourse of the linguistic variable y, R is the set of real numbers, X_p is the fuzzy set generated from the input linguistic variable x_p by a corresponding fuzzifier, and Y is the fuzzy set induced by the fuzzy inference engine. In the following, we discuss each of these components in more detail.

4.2 Fuzzification Interface

The *fuzzification interface* is a collection of fuzzifiers, each of which corresponds to an input linguistic variable and performs a mapping from a crisp number (real number) to a fuzzy set. Two types of these mappings commonly seen in the literature are:

(a) *Fuzzifier A* (F_A) – *Singleton Fuzzifier* : F_A is defined as a mapping from a crisp number, $z_0 \in \mathcal{Z}$, to a fuzzy singleton, $Z \in \mathcal{Z}$, where $\mathcal{Z}$ is a universe of discourse [104]. Formally,

$$\begin{cases} Z \triangleq F_A(z_0) \triangleq \{(z, \mu_Z(z)) \mid z \in \mathcal{Z}\}, \\ \mu_Z(z) \triangleq \begin{cases} 1, & if\ z = z_0 . \\ 0, & otherwise . \end{cases} \end{cases} \tag{4.1}$$

(b) *Fuzzifier B* (F_B) - General Fuzzifier : F_B is a mapping from a crisp number, $z_0 \in \mathcal{Z}$, to a fuzzy set, $Z \in \mathcal{Z}$,

$$Z \triangleq F_B(z_0) \triangleq \{(z, \mu_Z(z)) \mid z \in \mathcal{Z}\} . \tag{4.2}$$

where $\mu_Z(z_0) = 1$ and $\mu_Z(z)$ is some function defined subjectively which decreases from a maximum of 1 as z separates from z_0 [72].

A graphical illustration of these two fuzzifiers is shown in fig.4.2. Of these two types

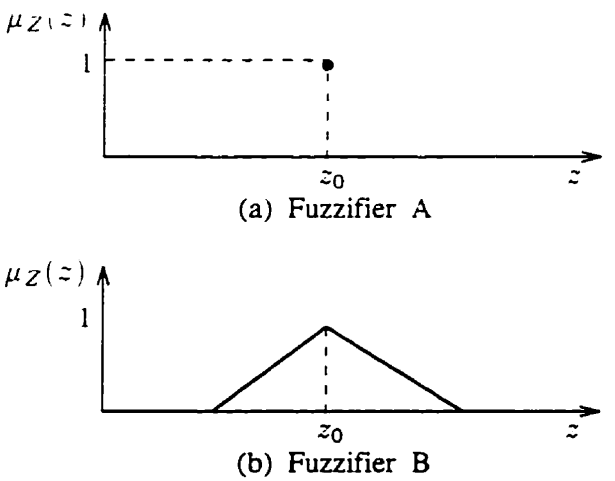


Figure 4.2: Illustration of Fuzzifiers

of fuzzifiers, only the singleton fuzzifier, F_A, is commonly used. The other type may be more useful for inputs corrupted by noise [184, Chap.2].

4.3 Fuzzy Rule Base

A fuzzy rule base is a repository of expert knowledge, which consists of a collection of fuzzy IF-THEN rules in the form,

$$C^i : IF\ (z_1\ is\ A_1^i\ and\ \ldots\ and\ z_P\ is\ A_P^i)\ THEN\ (y\ is\ B^i)\ . \tag{4.3}$$

where C^i, $i \in \{1, \ldots, I\}$, represents the ith fuzzy rule, I is the total number of fuzzy rules in the fuzzy rule base, z_p, $p = 1, \ldots, P$, are input linguistic variables in the universes of discourse $\mathcal{Z}_p$, and y is the output linguistic variable in the universe of discourse $\mathcal{Y}$.

For $p = 1, \ldots, P$ and $i = 1, \ldots, I$,

$$\begin{cases} A_p^i & \in & \{A_{pj_p} \mid A_{pj_p} \in \mathcal{Z}_p,\ j_p = 1, \ldots, J_p\}\ . \\ B^i & \in & \{B_k \mid B_k \in \mathcal{Y},\ k = 1, \ldots, K\}\ . \end{cases} \tag{4.4}$$

where A_{pj_p} are *primary fuzzy sets* defined in universes of discourse $\mathcal{Z}_p$, B_k are *primary fuzzy sets* defined in universe of discourse $\mathcal{Y}$, J_p and K represent the number of primary fuzzy sets defined in $\mathcal{Z}_p$ and $\mathcal{Y}$, respectively.

In this work, the term *primary fuzzy set* indicates those fuzzy sets defined to express linguistic IF-THEN rules, as distinguished from the fuzzy sets generated by fuzzifiers. In the IF part of expression (4.3), there are maximally $J_1 \times \cdots \times J_P$ different combinations of the primary fuzzy sets, $\{A_{pj_1}, \ldots, A_{pJ_p}\}$, which is the maximum possible number of IF-THEN rules in the fuzzy rule base.

The sentence connective "*and*" is usually implemented as a fuzzy conjunction, defined in eq.(3.20), in the Cartesian product space $\mathcal{Z}_1 \times \cdots \times \mathcal{Z}_P$, in which the underlying variables take values in different universes of discourse [88], i.e., the IF part of eq.(4.3) is a fuzzy relation $F^i \triangleq A_1^i \times \cdots \times A_P^i$ defined as

$$\begin{cases} F^i & \triangleq & \{((z_1, \ldots, z_P),\ \mu_{F^i}(z_1, \ldots, z_P)) \mid (z_1, \ldots, z_P) \in \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_P\}\ . \\ \mu_{F^i}(z_1, \ldots, z_P) & \triangleq & \mu_{A_1^i}(z_1) \star \cdots \star \mu_{A_P^i}(z_P)\ . \end{cases} \tag{4.5}$$

In control engineering, the most widely seen fuzzy conjunction operators for definition (4.5) are the "min" and algebraic product, i.e.,

$$\mu_{F^i}(z_1, \ldots, z_P) \triangleq \min(\mu_{A_1^i}(z_1), \ldots, \mu_{A_P^i}(z_P)) , \tag{4.6}$$

or

$$\mu_{F^i}(z_1, \ldots, z_P) \triangleq \mu_{A_1^i}(z_1) \cdot \mu_{A_2^i}(z_2) \cdots \mu_{A_P^i}(z_P) . \tag{4.7}$$

Each IF-THEN rule of expression (4.3) defines a fuzzy implication:

$$F^i \rightarrow B^i \triangleq \{((\mathbf{z}, y), \mu_{F^i \rightarrow B^i}(\mathbf{z}, y)) \mid (\mathbf{z}, y) \in \mathcal{Z} \times \mathcal{Y}\} , \tag{4.8}$$

where

$$\begin{cases} \mathbf{z} \triangleq \{z_1, \ldots, z_P\}^T , \\ \mathcal{Z} \triangleq \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_P . \end{cases} \tag{4.9}$$

The membership function, $\mu_{F^i \rightarrow B^i}(\mathbf{z}, y)$, is determined via eq.(3.20) in terms of $\mu_{F^i}(\mathbf{z})$ and $\mu_{B_i}(y)$, where the former is defined in (4.5).

In the literature, if the THEN part of an IF-THEN rule is defined as a linear combination of input variables, i.e.,

$$y \triangleq c_0 + c_1 z_1 + \ldots + c_P z_P , \tag{4.10}$$

where c_j, $j = 0, 1, \ldots, P$, are real parameters, the corresponding FLS is known as a FLS of the Takagi and Sugeno type [159]. As indicated in [184, Chap.1], an advantage of this kind of FLS is that it can provide a compact FLS equation, and parameter estimation and order determination methods can be used to estimate the parameters c_i and the order I. Its weakness is that the THEN part of the rule is not fuzzy, and it does not, therefore, provide a natural framework for incorporation of fuzzy rules from human experts. This type of FLS will not be considered in this work.

4.4 Fuzzy Inference Engine

The fuzzy inference engine is a fuzzy logic based decision making scheme which, by combining the IF-THEN rules in the fuzzy rule base in certain manner, induces output fuzzy sets in $\mathcal{Y}$ from input fuzzy sets in $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_P$.

With a input fuzzy set $Z \in \mathcal{Z}$, where

$$\begin{cases} Z & \triangleq Z_1 \times \cdots \times Z_P \triangleq \{(\mathbf{z},\ \mu_Z(\mathbf{z})) \mid (\mathbf{z}) \in \mathcal{Z}\}\ , \\ \mu_Z(\mathbf{z}) & \triangleq \mu_{Z_1}(z_1) \star \cdots \star \mu_{Z_P}(z_P)\ , \end{cases} \tag{4.11}$$

and by using the sup-star compositional rule of inference, eq.(3.21), each individual IF-THEN rule, C^i, induces a component, denoted as Y^i, of the output fuzzy set $Y \in \mathcal{Y}$, i.e., for $i = 1, \ldots, I$,

$$\begin{cases} Y^i & \triangleq Z \circ (F^i \rightarrow B^i) = \{(y,\ \mu_{Y^i}(y)) \mid y \in \mathcal{Y}\}\ , \\ \mu_{Y^i}(y) & \triangleq \mu_{Z \circ (F^i - B^i)}(y) = \sup_{\mathbf{z} \in \mathcal{Z}} \{\mu_Z(\mathbf{z}) \star \mu_{F^i - B^i}(\mathbf{z}, y)\}\ . \end{cases} \tag{4.12}$$

The membership function of the fuzzy implication, $\mu_{F^i - B^i}(\mathbf{z}, y)$, is defined in eq.(3.20) in terms of $\mu_{F^i}(\mathbf{z})$ and $\mu_{B^i}(y)$, and the membership functions for the fuzzy relations $\mu_{F^i}(\mathbf{z})$ and $\mu_Z(\mathbf{z})$ are defined in eqs. (4.5) and (4.11), respectively.

The output fuzzy set Y is the collection of the I components, Y^i, $i = 1, \ldots, I$, induced by the IF-THEN rules in the fuzzy rule base, i.e.,

$$Y \triangleq \{Y^i \mid Y^i \in \mathcal{Y},\ i = 1, \ldots, I\}\ . \tag{4.13}$$

4.5 Defuzzification Interface

The *defuzzification interface* is a collection of defuzzifiers, or a single defuzzifier in the case of single output, each of which corresponds to an output linguistic variable and performs a mapping from a fuzzy set, Y, to a crisp number, y. There are three kinds of defuzzifiers commonly seen in the literature.

(a) The first type, called the *Least-Maximum Method* (LMM) [206], is defined as

$$y_a \triangleq D_a(Y) \triangleq \min_{y_m \in \mathcal{Y}} \{| y_m |,\ m = 1, \ldots, M\} , \tag{4.14}$$

where y_m, $m = 1, \ldots, M$, are the supports in fuzzy set Y that have the (same) maximum grades of membership, i.e.

$$y_m \in \{y \mid y = arg \sup_{y \in \mathcal{Y}} \{\mu_Y(y)\}\} , \quad m = 1, \ldots, M . \tag{4.15}$$

The LMM defuzzifier generates a crisp number, y_a, which is the support point that has the maximum grade of membership in the fuzzy set, Y, or, in case there is more than one support corresponding to the same maximum grade of membership, i.e., $N > 1$, y_a takes the support that has the least absolute value.

(b) The second type, called the *Mean of Maximum Method* (MOM) [104], is defined as

$$y_b \triangleq D_b(Y) \triangleq \frac{1}{M} \sum_{m=1}^{M} y_m , \tag{4.16}$$

where y_b is the mean of the supports y_m, $m = 1, \ldots, M$. If $M = 1$, this type of defuzzifier generates the same crisp number as that of the first type defuzzifier.

(c) The third type, called *Centroid Method* (CM) [21], is defined as

$$y_c \triangleq D_c(Y) \triangleq \frac{\sum_{i=1}^{I} \mu_{Y^i}(\overline{y}^i) \cdot \overline{y}^i}{\sum_{i=1}^{I} \mu_{Y^i}(\overline{y}^i)} , \tag{4.17}$$

where $\overline{y}^i$, $i = 1, 2, \ldots, I$, are the centers of the primary fuzzy sets, B^i, i.e.,

$$\overline{y}^i \triangleq arg \sup_{y \in \mathcal{Y}} \{\mu_{B^i}(y)\} . \tag{4.18}$$

and $\mu_{Y^i}(y)$ can be obtained with eq.(4.12). y_c represents the centroid of the supports $\overline{y}^i$, which is the reason that this approach is called the *Centroid Method*.

The CM is the most popular defuzzification strategy, as it is said to yield generally smoother responses and better steady state performance than the MOM [147]. Similar observations were made in an aircraft flight control study [86], where it is found that although both the MOM and CM yielded satisfactory performance, the results with the CM were superior.

The MOM demonstrated better performance than the LMM in certain experiments [75] and frequently yielded better transient performance than the CM [147]. The LMM strategy, unlike the other two strategies, has not been seen much in practical applications.

Throughout this work, the centroid defuzzification method will be used, because its expression is mathematically more tractable than the others.

4.6 Overall Mappings of Fuzzy Logic Systems

We have seen that there are many possible combinations of fuzzifiers, defuzzifiers, fuzzy implication operations and fuzzy inference mechanisms, which result in many different fuzzy logic system configurations, or overall mappings. This provides a great deal of freedom of choices of a specific FLS configuration which fits the constraint and requirement of a particular task.

Among the two fuzzifiers mentioned in section 4.2, the singleton fuzzifier produces the simplest FLS, and only this kind of fuzzifiers will be used in this work. Formally, for the FLS shown in fig.4.1, $p = 1, \ldots, P$,

$$\begin{cases} Z_p & \triangleq \{(z_p,\ \mu_{Z_p}(z_p)) \mid z_p \in \mathcal{Z}_p\}\ . \\ \mu_{Z_p}(z_p) & \triangleq \begin{cases} 1, & if\ z_p = z_{p0}\ . \\ 0, & otherwise\ . \end{cases} \end{cases} \tag{4.19}$$

where z_{p0} is a crisp value of the variable z_p.

In the following, we derive two overall mappings which are particularly useful in later development of this work:

4.6.1 FLS-I

- "min" operation for fuzzy relation:

  Equation (4.5) is written as

$$\begin{cases} F^i & \triangleq \{(\mathbf{z},\ \mu_{F^i}(\mathbf{z})) \mid \mathbf{z} \in \mathcal{Z}\} \ . \\ \mu_{F^i}(\mathbf{z}) & \triangleq \min\{\mu_{A_1^i}(z_1),\ \ldots.\ \mu_{A_P^i}(z_P)\} \ . \end{cases} \tag{4.20}$$

  where $\mathbf{z} = \{z_1, \ldots, z_P\}^T$, $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_P$, $A_p^i \in \mathcal{Z}_p$ are primary fuzzy sets.

- "min" operation for fuzzy implication:

  Equation (4.8) is written as

$$\begin{cases} F^i \rightarrow B^i & \triangleq \{((\mathbf{z}, y),\ \mu_{F^i \rightarrow B^i}(\mathbf{z}, y)) \mid (\mathbf{z}, y) \in \mathcal{Z} \times \mathcal{Y}\} \ . \\ \mu_{F^i \rightarrow B^i}(\mathbf{z}, y)) & \triangleq \min\{\mu_{F^i}(\mathbf{z}),\ \mu_{B^i}(y)\} \ . \end{cases} \tag{4.21}$$

  where B^i are primary fuzzy sets in $\mathcal{Y}$.

- "sup-min" compositional rule of fuzzy inference:

  Equation (4.12) is written as

$$\begin{cases} Y^i & \triangleq Z \circ (F^i \rightarrow B^i) \triangleq \{(y,\ \mu_{Y^i}(y)) \mid y \in \mathcal{Y}\} \ . \\ \mu_{Y^i}(y) & \triangleq \sup_{\mathbf{z} \in \mathcal{Z}}\{\min\{\mu_Z(\mathbf{z}),\ \mu_{F^i \rightarrow B^i}(\mathbf{z}, y)\}\} \ . \end{cases} \tag{4.22}$$

  Use of eq.(4.21) in eq.(4.22) yields

$$\mu_{Y^i}(y) = \sup_{\mathbf{z} \in \mathcal{Z}}\{\min\{\mu_Z(\mathbf{z}),\ \mu_{F^i}(\mathbf{z}),\ \mu_{B^i}(y)\}\} \ . \tag{4.23}$$

  Since we use the "min" operation for fuzzy relations, the membership function $\mu_Z(\mathbf{z})$ is expressed as

$$\mu_Z(\mathbf{z}) \triangleq \min\{\mu_{Z_1}(z_1),\ \ldots.\ \mu_{Z_P}(z_P)\} \ . \tag{4.24}$$

In view of eqs. (4.20) and (4.24), eq.(4.23) can be rewritten as

$$\mu_{Y^i}(y) = \sup_{\mathbf{z}\in\mathcal{Z}}\{\min\{\mu_{Z_1}(z_1),\ldots,\mu_{Z_P}(z_P),\mu_{A_1^i}(z_1),\ldots,\mu_{A_P^i}(z_P),\mu_{B^i}(y)\}\} . \tag{4.25}$$

Given a crisp input, $\mathbf{z}_0 \triangleq \{z_{10}, \ldots, z_{P0}\}^T$, use of eq.(4.19) in eq.(4.25) results in

$$\mu_{Y^i}(y) = \min\{\mu_{A_1^i}(z_{10}), \ldots, \mu_{A_P^i}(z_{P0}), \mu_{B^i}(y)\} . \tag{4.26}$$

- Using the CM defuzzifier, eq.(4.17), the crisp output signal of the FLS is

$$y_c \triangleq \frac{\sum_{i=1}^{I}\mu_{Y^i}(\overline{y}^i)\cdot\overline{y}^i}{\sum_{i=1}^{I}\mu_{Y^i}(\overline{y}^i)} . \tag{4.27}$$

Here, $\overline{y}^i$, $i = 1, 2, \ldots, I$, are defined as [88, 91]

$$\overline{y}^i \triangleq \min_{y^i\in Y^i}\{\,|y^i| : y^i = arg\sup_{y\in\mathcal{Y}}\{\mu_{Y^i}(y)\}\,\} . \tag{4.28}$$

This will be explained in more detail in chapter 6 through examples.

The FLS specified above is referred to as FLS-I in this work.

4.6.2 FLS-II

- Algebraic product operation for fuzzy relation:

Equation (4.5) is written as

$$\begin{cases} F^i & \triangleq \{(\mathbf{z}, \mu_{F^i}(\mathbf{z})) \mid \mathbf{z}\in\mathcal{Z}\} , \\ \mu_{F^i}(\mathbf{z}) & \triangleq \mu_{A_1^i}(z_1)\cdot\mu_{A_2^i}(z_2)\cdots\mu_{A_P^i}(z_P) , \end{cases} \tag{4.29}$$

where $\mathbf{z} = \{z_1,\ldots,z_P\}^T$, $\mathcal{Z} = \mathcal{Z}_1\times\cdots\times\mathcal{Z}_P$, $A_p^i \in \mathcal{Z}_p$ are primary fuzzy sets, and "$\cdot$" represents the algebraic product operation.

- Algebraic product operation for fuzzy implication:

  Equation (4.8) is written as

$$\begin{cases} F^i \rightarrow B^i & \triangleq \{((\mathbf{z}, y),\ \mu_{F^i \rightarrow B^i}(\mathbf{z}, y)) \mid (\mathbf{z}, y) \in \mathcal{Z} \times \mathcal{Y}\} \ . \\ \mu_{F^i \rightarrow B^i}(\mathbf{z}, y) & \triangleq \mu_{F^i}(\mathbf{z}) \cdot \mu_{B^i}(y) \ , \end{cases} \tag{4.30}$$

  where B^i are primary fuzzy sets in $\mathcal{Y}$.

- "sup-product" compositional rule of fuzzy inference:

  Equation (4.12) is written as

$$\begin{cases} Y^i & \triangleq Z \circ (F^i \rightarrow B^i) \triangleq \{(y,\ \mu_{Y^i}(y)) \mid y \in \mathcal{Y}\} \ . \\ \mu_{Y^i}(y) & \triangleq \sup_{\mathbf{z} \in \mathcal{Z}} \{\mu_Z(\mathbf{z}) \cdot \mu_{F^i \rightarrow B^i}(\mathbf{z}, y)\} \ . \end{cases} \tag{4.31}$$

  Using eq.(4.30) in eq.(4.31) yields

$$\mu_{Y^i}(y) = \sup_{\mathbf{z} \in \mathcal{Z}} \{\mu_Z(\mathbf{z}) \cdot \mu_{F^i}(\mathbf{z}) \cdot \mu_{B^i}(y)\} \ . \tag{4.32}$$

  Since we use the algebraic product operation for fuzzy relations, the membership function $\mu_Z(\mathbf{z})$ is expressed as

$$\mu_Z(\mathbf{z}) \triangleq \mu_{Z_1}(z_1) \cdot \mu_{Z_2}(z_2) \cdots \mu_{Z_P}(z_P) \ . \tag{4.33}$$

  In view of eqs. (4.29) and (4.33), eq.(4.32) can be rewritten as

$$\mu_{Y^i}(y) = \sup_{\mathbf{z} \in \mathcal{Z}} \{[\prod_{p=1}^{P} \mu_{Z_p}(z_p)] \cdot [\prod_{p=1}^{P} \mu_{A_p^i}(z_p)] \cdot \mu_{B^i}(y)\} \ . \tag{4.34}$$

  For a crisp input, $\mathbf{z}_0 \triangleq \{z_{10}, \ldots, z_{P0}\}^T$, use of eq.(4.19) in eq.(4.34) results in

$$\mu_{Y^i}(y) = \begin{cases} \mu_{A_1^i}(z_{10}) \cdot \mu_{A_2^i}(z_{20}) \cdots \mu_{A_P^i}(z_{P0}) \cdot \mu_{B^i}(y)\} \ , & if\ \mathbf{z} = \mathbf{z}_0 \ . \\ 0. & otherwise. \end{cases} \tag{4.35}$$

- Using the CM defuzzifier, eqs.(4.17–4.18), the crisp output signal, y, of the FLS is

$$y_c \triangleq \frac{\sum_{i=1}^{I} \mu_{Y^i}(\overline{y}^i) \cdot \overline{y}^i}{\sum_{i=1}^{I} \mu_{Y^i}(\overline{y}^i)} \ . \tag{4.36}$$

$$\overline{y}^i \triangleq arg \sup_{y \in \mathcal{Y}} \{\mu_{B^i}(y)\} \ . \tag{4.37}$$

For simplicity, it may be assumed that

$$\mu_{B^l}(\overline{y}^l) \equiv 1 \ . \tag{4.38}$$

Use of eq.(4.35) in eq.(4.36) and considering eq.(4.38) yield

$$y = \frac{\sum_{l=1}^{I} \overline{y}^l \prod_{p=1}^{P} \mu_{A_p^l}(z_p)}{\sum_{l=1}^{I} \prod_{p=1}^{P} \mu_{A_p^l}(z_p)} \ . \tag{4.39}$$

where, for simplicity, y_c is abbreviated as y, and z_{p0} is abbreviated as z_p.

The FLS which satisfies these conditions is hereafter referred to as FLS-II in this work.

4.6.3 Membership Functions and an Overall FLS Mapping

In fuzzy logic systems, it is necessary to specify membership functions for primary fuzzy sets. In practice, two types of membership functions have been popular, namely the Gaussian and triangular type membership functions.

For a given linguistic variable, u, in universe of discourse, $\mathcal{U}$, a Gaussian type membership function of u in a fuzzy set $U^l \in \mathcal{U}$ is expressed as

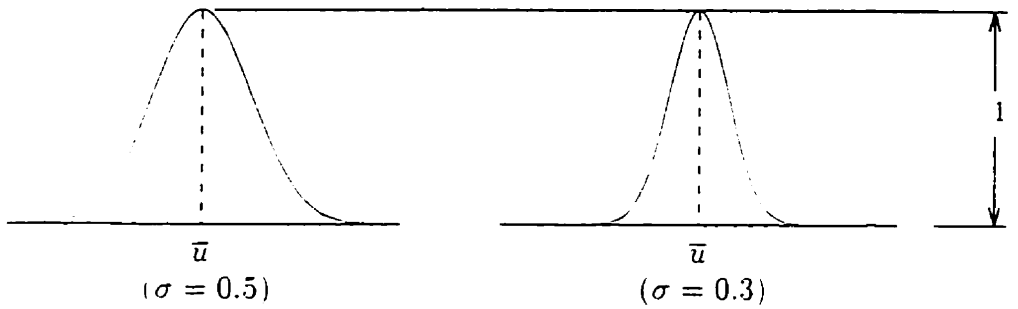
$$\mu_{U^l}(u) = \exp[-\frac{1}{2}(\frac{u - \overline{u}}{\sigma})^2] \ . \tag{4.40}$$

where $\overline{u}$ is the center of the fuzzy set U^l and σ is a parameter that characterizes the shape of the membership function. Lower values of σ produce narrowing of the function, as shown in fig.4.3(a).

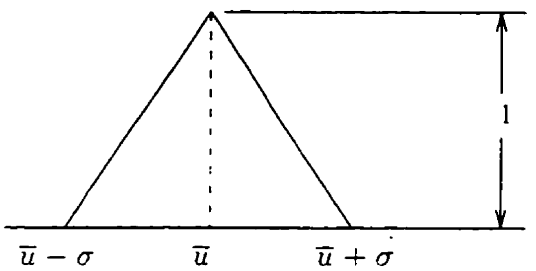
The triangular membership function is expressed as

$$\mu_{U^l}(u) = \begin{cases} 1 - \left|\frac{u-\overline{u}}{\sigma}\right|, & if \quad u \in [\overline{u} - \sigma, \overline{u} + \sigma] \ , \\ 0 \ . & otherwise \ . \end{cases} \tag{4.41}$$

where $\overline{u}$ is the center of the fuzzy set U^l, and σ controls the shape of the function, as shown in fig.4.3(b).



(a) Gaussian Type Membership Functions



(b) Triangular Type Membership Function

Figure 4.3: Gaussian and Triangular Membership Functions

The Gaussian type membership function will be used throughout this work for its compact mathematical expression and ease of manipulation.

With a Gaussian type membership function, eq.(4.40), in FLS-II, eq.(4.39) becomes

$$y \triangleq f(\mathbf{z}) \triangleq \frac{\sum_{i=1}^{I} \bar{y}^i \prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{z_p - \bar{z}_p^i}{\sigma_p^i})^2]}{\sum_{i=1}^{I} \prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{z_p - \bar{z}_p^i}{\sigma_p^i})^2]} . \tag{4.42}$$

It is shown in [184] that for any given real continuous function g on a compact set U, there exists a fuzzy logic system f in the form of eq.(4.42), such that g can be approximated by f over U to any desired degree of accuracy.

We simplify our notation by defining

$$\theta_i(\mathbf{z}) \triangleq \frac{\prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{z_p - \bar{z}_p^i}{\sigma_p^i})^2]}{\sum_{i=1}^{I} \prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{z_p - \bar{z}_p^i}{\sigma_p^i})^2]} . \tag{4.43}$$

which is called a *fuzzy basis function* in the literature [179]. Let

$$\begin{cases} \boldsymbol{\Theta}(\mathbf{z}) & \triangleq & \{\theta_1(\mathbf{z}), \ldots, \theta_I(\mathbf{z})\}^T \ . \\ \overline{\mathbf{Y}} & \triangleq & \{\overline{y}_1, \ldots, \overline{y}_I\}^T \ . \end{cases} \tag{4.44}$$

Equation (4.42) can then be rewritten as

$$y = \sum_{i=1}^{I} \overline{y}_i \cdot \theta_i(\mathbf{z}) = \boldsymbol{\Theta}^T(\mathbf{z})\overline{\mathbf{Y}} \ . \tag{4.45}$$

In eq.(4.43), it is clear that $\boldsymbol{\Theta}$ is characterized by the number of primary fuzzy sets, J_p, in each universe of discourse $\mathcal{Z}_p$, $p = 1, \ldots, P$, as well as the position $\overline{z}_p^i$ and shape σ_p^i of the membership function of each primary fuzzy set, where

$$\begin{cases} \overline{z}_p^i & \in & \{\overline{z}_{pj_p} \mid \overline{z}_{pj_p} \in A_{pj_p}, \ j_p = 1, \ldots, J_p\} \ . \\ \sigma_p^i & \in & \{\sigma_{pj_p} \mid j_p = 1, \ldots, J_p\} \ . \end{cases} \tag{4.46}$$

and $\overline{z}_{pj_p}$ and σ_{pj_p} are centers and shape parameters of the primary fuzzy sets, A_{pj_p}, respectively. If we fix these parameters and leave only $\overline{y}_i$, $i = 1, \ldots, I$, as adjustable parameters, then expression (4.45) is linear in its parameters.

Chapter 5

More On Fuzzy Logic Systems

5.1 Quantitative Measures of IF-THEN Rule Performance

5.1.1 Introduction

The fuzzy rule base, which consists of IF-THEN rules, is the heart of a fuzzy logic system in the sense that all other components of the FLS are used to interpret these rules and make the rules usable for specific problems [184]. Difficulties with logical properties of the rules, their completeness, consistency, etc., may lead to unsatisfactory results [129]. These problems may be caused either by poor initial design or because of changing operating environment and system parameters. Therefore, it is desirable to monitor the performance of the IF-THEN rules on-line and warn system operators of any problems so that corrective measures can be made in time, and satisfactory system performance maintained.

In the following, we define quantitative criteria that can reveal certain problems with the rules, and demonstrate their application through numerical examples.

5.1.2 Quantitative Measures

The completeness of a fuzzy rule base is defined by Pedrycz [129] as

Definition 5.1 *Completeness of Fuzzy Rule Base:*
Consider a fuzzy rule base consisting of IF-THEN rules of the form of expression (4.3) and of size I. It is said to be complete, if

$$\forall\, \mathbf{z} \in \mathcal{Z},\ \exists\, i \in [1,\ I],\ \text{such that } \mu_{F^i}(\mathbf{z}) \geq \varepsilon\ ,\quad \varepsilon \in (0,1]\ . \tag{5.1}$$

where $\mathbf{z} = \{z_1, \ldots, z_P\}^T$ is a crisp input vector, F^i, defined in eq.(4.5), is a fuzzy relation in the IF part of the ith IF-THEN rule, and ε is a subjective positive constant.

We now introduce a quantitative index that indicates the degree of completeness of a fuzzy rule base. It should be kept in mind that as this quantitative index is for on-line applications, it should be as simple as possible and not generate a significant extra computational burden.

Definition 5.2 *Completeness Index (CI):*
For a fuzzy rule base consisting of IF-THEN rules of the form of expression (4.3) and of size I, the completeness index for the ith rule is defined as

$$CI^i(\mathbf{z}) \triangleq \mu_{F^i}(\mathbf{z})\ . \tag{5.2}$$

where $\mathbf{z} = \{z_1, \ldots, z_P\}^T \in \mathcal{Z}$ is a crisp input vector, and F^i, defined in eq.(4.5), is a fuzzy relation in the IF part of the ith IF-THEN rule.

We now apply the completeness index to test the completeness of fuzzy rule base.

Lemma 5.1 *A fuzzy rule base of size I is incomplete if for $i = 1, \ldots, I$, and $\forall\, \mathbf{z} \in \mathcal{Z}$,*

$$CI^i(\mathbf{z}) < \varepsilon, \qquad \varepsilon \in (0,1], \tag{5.3}$$

where ε is a subjective positive constant.

Proof: Direct from the definition of completeness of fuzzy rule bases and the definition of the completeness index.

Remarks:

- The completeness of a fuzzy rule base, i.e., eq.(5.1), means that [130, Chap.6]

  (i) the primary fuzzy sets defined in the input universes of discourse overlap;

  (ii) each primary fuzzy set is used in at least one rule.

  Comparing eq.(5.3) with eq.(5.1), it is clear that the index CI can indicate potential problems in these aspects.

- Since CI is defined as μ_{F^i}, it does not generate much additional computational burden and is therefore suitable for on-line applications. The only extra calculation required is its comparison with the predefined threshold parameter ε.

Definition 5.3 *Fire-Density Index (FDI):*
Consider a fuzzy rule base of size I triggered M times in certain period of time, $[t_1, t_2]$. Then the arithmetic average of the completeness index of the ith rule is defined as the Fire-Density Index in $[t_1, t_2]$. Formally,

$$FDI^i\,|_{t_1}^{t_2} \triangleq \frac{1}{M}\sum_{m=1}^{M} CI_m^i(\mathbf{z}) \ . \tag{5.4}$$

The index FDI^i, $i = 1, \ldots, I$, indicates the share of usage of individual rules. If FDI^i distributes very unevenly with respect to i over a designated period of time, e.g., as shown in fig.5.1, it means that some rules are heavily used, such as the fifth and sixth rules, while others, such as the first and tenth, are little used. This indicates that the input universes of discourse may have been ill-partitioned, and there is at least a room to improve the fuzzy partition.

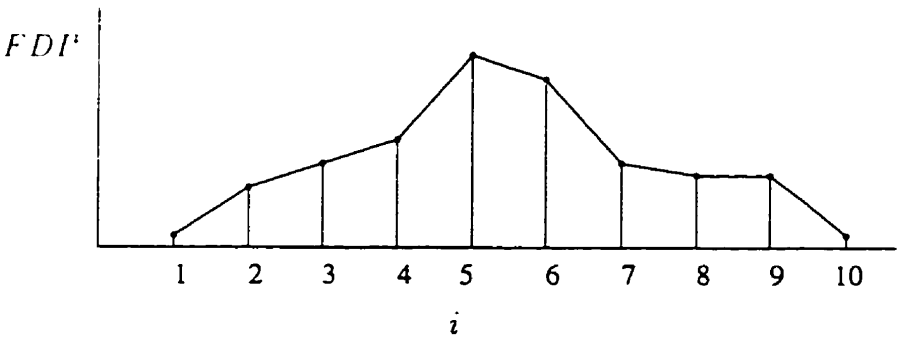


Figure 5.1: Distribution of Fire-Density Indices

5.1.3 Numerical Examples

Example 1. Completeness of Fuzzy Rule Base

Consider a linguistic variable, $u \in \mathcal{U} \triangleq [-3, 3]$, whose values are random numbers normally distributed with mean 0 and variance 1. Its density function is [124]

$$f(u) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{u^2}{2\sigma^2}) \ . \tag{5.5}$$

which is graphically illustrated in fig.5.2(a). Define five primary fuzzy sets, $\{N2, N1, ZR, P1, P2\} \in \mathcal{U}$, whose membership functions are Gaussian type membership functions, (eq.(4.40)), centered at {-2, -1, 0, 1, 2} with shape parameters, $\sigma \triangleq 0.3$, for all the primary fuzzy sets, as shown in fig.5.2(b). The corresponding fuzzy rule base we assign to have five IF-THEN rules as shown in table 5.1. With the desired threshold parameter, ε, set at 0.05, i.e.

$$\varepsilon \triangleq 0.05 \ . \tag{5.6}$$

for crisp input $u = 2.735 \in \mathcal{U}$,

$$\begin{cases} CI^1(2.735) = 0 < \varepsilon; \\ CI^2(2.735) = 0 < \varepsilon; \\ CI^3(2.735) = 0 < \varepsilon; \\ CI^4(2.735) = 0 < \varepsilon; \\ CI^5(2.735) = 0.0497 < \varepsilon. \end{cases} \tag{5.7}$$

Table 5.1: IF-THEN Rules for Linguistic Variable u

C^1: IF (u is N2) THEN ...:
C^2: IF (u is N1) THEN ...;
C^3: IF (u is ZR) THEN ...:
C^4: IF (u is P1) THEN ...;
C^5: IF (u is P2) THEN ....

Therefore, according to lemma 5.1, this fuzzy rule base is incomplete. For this simple example, it is obvious that the fuzzy partition must be improved. We may redefine the membership functions for the fuzzy sets "N2" and "P2" as,

$$\mu_{N2}(u) \triangleq \begin{cases} 1. & if\ u < -2: \\ \exp[-\frac{1}{2}(\frac{u+2}{0.3})^2]. & if\ u \geq -2. \end{cases} \tag{5.8}$$

$$\mu_{P2}(u) \triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{u-2}{0.3})^2]. & if\ u < 2: \\ 1. & if\ u \geq 2. \end{cases} \tag{5.9}$$

The new primary fuzzy sets are shown in fig.5.2(c), and the corresponding completeness indices for $u = 2.735$ are

$$\begin{cases} CI^1(2.735) = 0 < \varepsilon: \\ CI^2(2.735) = 0 < \varepsilon: \\ CI^3(2.735) = 0 < \varepsilon: \\ CI^4(2.735) = 0 < \varepsilon: \\ CI^5(2.735) = 1 > \varepsilon. \end{cases} \tag{5.10}$$

It is clear that the problem of incompleteness of the fuzzy rule base has been corrected.

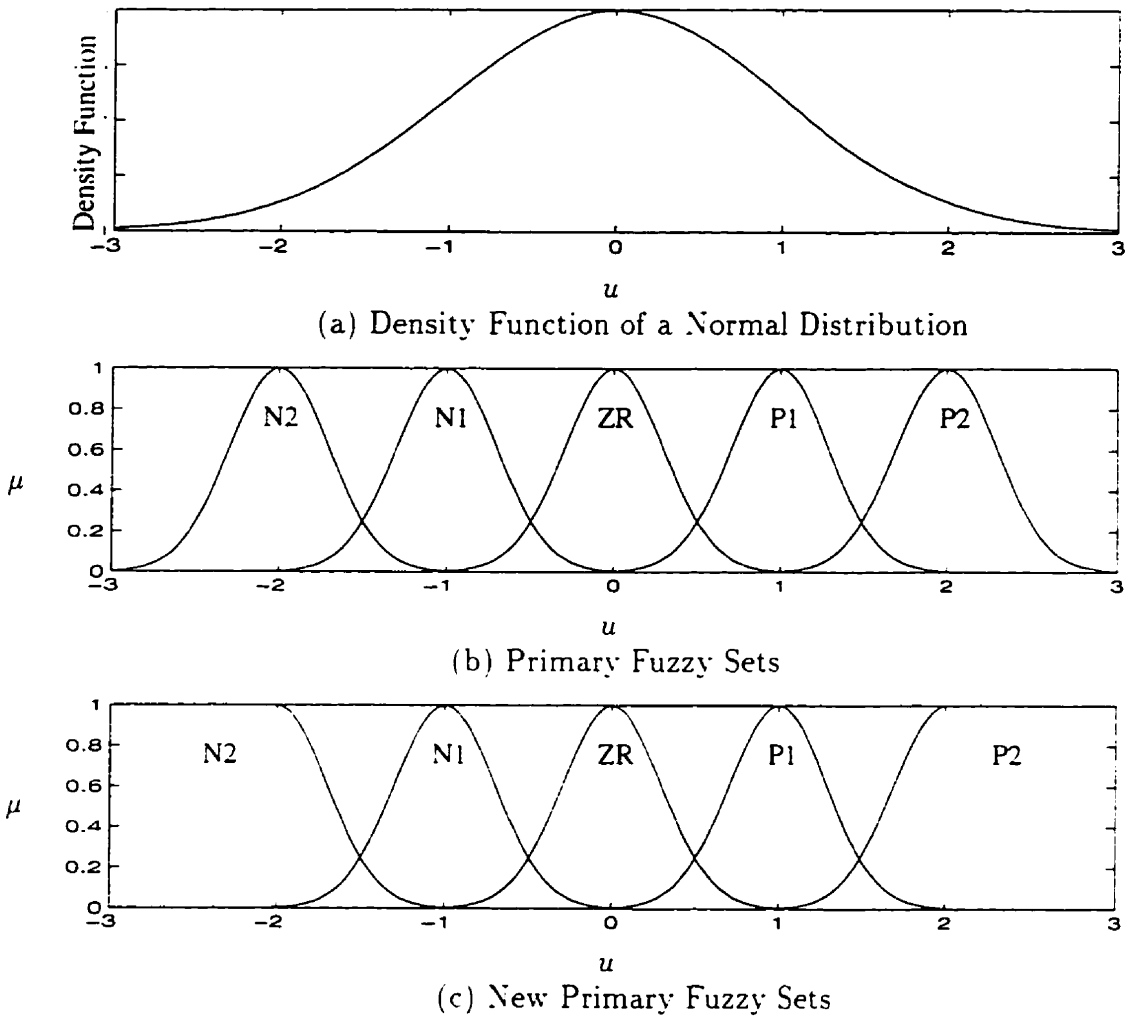


Figure 5.2: Completeness of Fuzzy Rule Base

Example 2. Distribution of Fire-Density Index

Consider the same linguistic variable $u \in \mathcal{U} \triangleq [-3, 3]$ as in example 1, whose density function is plotted in fig.5.3(a) again. Define five primary fuzzy sets, {N2, N1, ZR, P1, P2}$\in \mathcal{U}$, whose membership functions are Gaussian, (eq.(4.40)), centered at {-2, -1, 0, 1, 2} with shape parameter $\sigma \triangleq 0.35$ for all the primary fuzzy sets, as shown in fig.5.3(b), where

$$\mu_{N2}(u) \triangleq \begin{cases} 1. & if\ u < -2: \\ \exp[-\frac{1}{2}(\frac{u+2}{0.35})^2] , & if\ u \geq -2. \end{cases} \tag{5.11}$$

$$\mu_{P2}(u) \triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{u-2}{0.35})^2] , & if\ u < 2: \\ 1. & if\ u \geq 2. \end{cases} \tag{5.12}$$

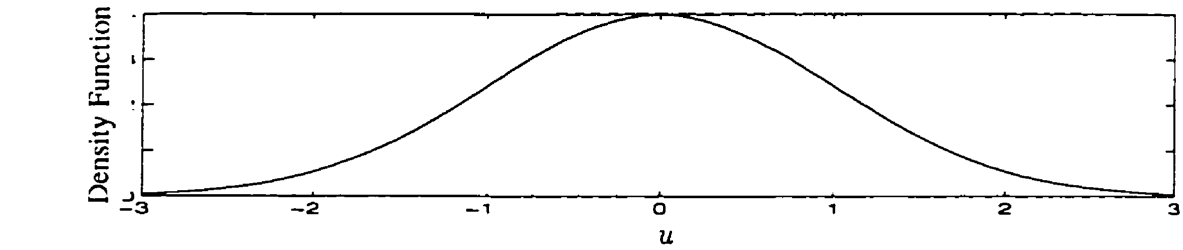
The corresponding fuzzy rule base has five IF-THEN rules, the same as those in table 5.1. We take 2000 samples of variable u. Correspondingly, the fuzzy rule base is triggered $M = 2000$ times. The fire density indices for the five rules over the 2000 samples, are shown in fig.5.3(c), where FDI^2, FDI^3 and FDI^4 are significantly larger than FDI^1 and FDI^5. This uneven share of the IF-THEN rules is due to the fact that samples are not evenly distributed, and most of the samples are in [-1, 1], where the density function of variable u has high values.

We can define more primary fuzzy sets in the region where the density function is large, and define fewer primary fuzzy sets where the density function has lower values, which is intuitively more appealing. We now define seven primary fuzzy sets, {N3, N2, N1, ZR, P1, P2, P3}$\in \mathcal{U}$, whose membership functions are Gaussian centered at {-1.5, -1.0, -0.5, 0, 0.5, 1.0, 1.5} with corresponding shape parameters, σ, being {0.35, 0.3, 0.25, 0.25, 0.25, 0.3, 0.35}, as shown in fig.5.3(d), where

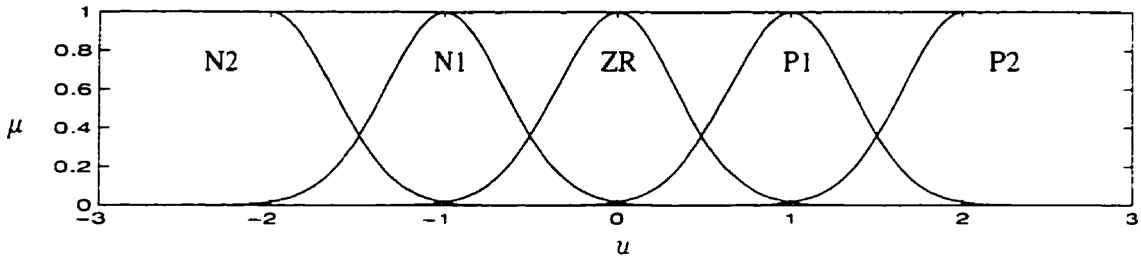
$$\mu_{N3}(u) \triangleq \begin{cases} 1. & if\ u < -1.5: \\ \exp[-\frac{1}{2}(\frac{u+1.5}{0.35})^2] , & if\ u \geq -1.5. \end{cases} \tag{5.13}$$

$$\mu_{P3}(u) \triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{u-1.5}{0.35})^2] , & if\ u < 1.5: \\ 1. & if\ u \geq 1.5. \end{cases} \tag{5.14}$$

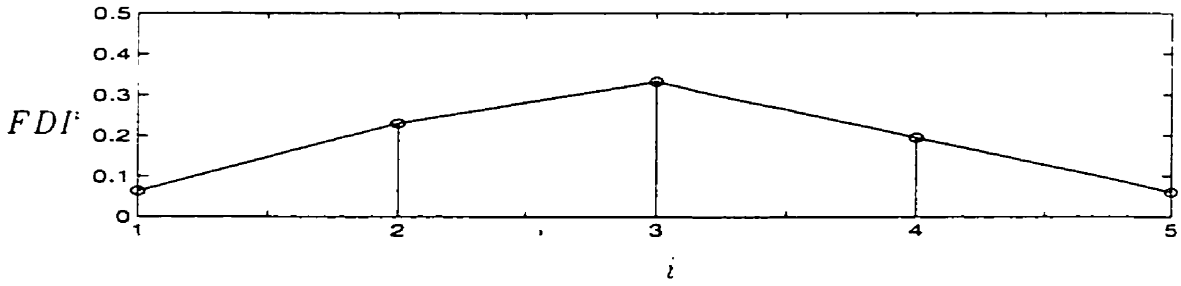
The corresponding fuzzy rule base has seven IF-THEN rules, which are similar to those in table 5.1. Using the same 2000 samples mentioned above, the fire-density indices for the seven rules are shown in fig.5.3(e), where FDI^i, $i = 1, \ldots, 7$, are quite evenly distributed.



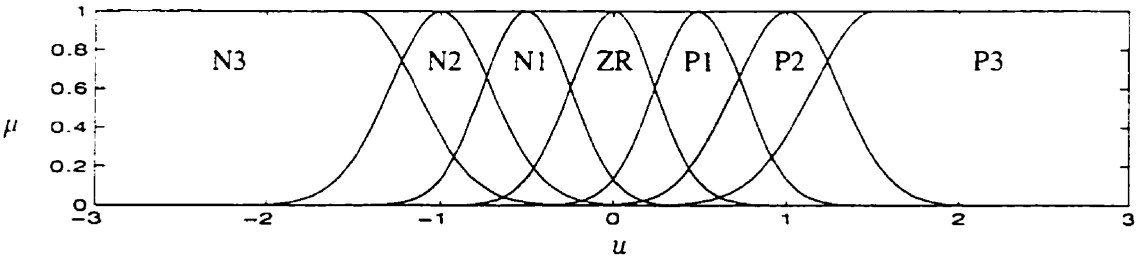
(a) Density Function of a Normal Distribution



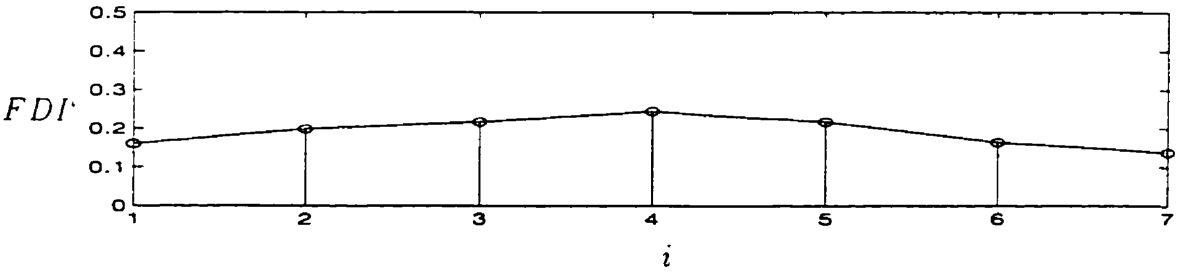
(b) Primary Fuzzy Sets



(c) FDI^i Distribution



(d) New Primary Fuzzy Sets



(e) New FDI^i Distribution

Figure 5.3: Data Distribution, Fuzzy Partition and Fire-Density Index

5.1.4 Concluding Remarks

- Two quantitative measures, completeness index and fire density index, for IF-THEN rule performance are proposed here. They can indicate certain difficulties in a particular fuzzy rule base, such as incompleteness and uneven share of usage of individual rules, and these indices were illustrated via numerical examples.

- With these indices integrated into a fuzzy rule base, the reliability of the corresponding fuzzy logic system is likely to be enhanced, since system operators can be warned of certain weaknesses of the FLS and take remedial measures in time.

- These indices are obtained with little additional computational burden, since CI^t is defined as μ_{F^t}, FDI^t is defined as the arithmetic average of CI^t over a certain period of time.

- A full understanding of these indices, as well as their values for system performance are yet to be investigated. There is no solid theory available at present, and their merits must therefore be understood and explained based on experience and with regard to specific problems. In addition, there is no automatic means available to correct the problems they uncover - another subject for further investigation.

5.2 Statistical Confidence Measure of Fuzzy Logic System Performance

When a fuzzy logic system is used in function approximation problems, it is desirable to know both the approximation accuracy of the FLS approximators and the quality of performance of the individual IF-THEN rules. In this section, by applying interval estimation techniques of statistics, we develop a statistical index which can be used to address this. Specifically, we present statistical estimates of confidence limits on both the overall FLS outputs and predictions of individual IF-THEN rules.

5.2.1 Interval Estimation Problem

The interval estimation of a parameter η is an interval (η_1, η_2), the endpoints of which are functions of an observation vector $\mathbf{x}$, i.e., $\eta_1 = g_1(\mathbf{x})$ and $\eta_2 = g_2(\mathbf{x})$. We shall say that (η_1, η_2) is a γ confidence interval of η, if the probability that $\eta \in (\eta_1, \eta_2)$ is γ [124, Chap.9], i.e.,

$$P\{\eta_1 < \eta < \eta_2\} = \gamma \ . \tag{5.15}$$

The constant γ is the confidence coefficient of the estimate and thus is a subjective measure of our confidence that the unknown η is in the interval (η_1, η_2). The estimate is expected to be correct in 100γ percent of cases. If γ is close to 1, we can expect with near certainty that η is in (η_1, η_2), but the difference, $\eta_2 - \eta_1$, is usually large. With γ reduced, $\eta_2 - \eta_1$ is reduced but the estimate is less reliable. It is clear that, for given γ, smaller $\eta_2 - \eta_1$ indicates better performance of the system. The objective of interval estimation is to determine the functions $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ for given γ.

5.2.2 Confidence Interval for the Mean of FLS Approximation Errors

For a function approximation problem, assume we have M input-output data pairs, $\{(\mathbf{z}_m, y_m^d) \mid m = 1, \ldots, M\}$, where $\mathbf{z}_m \triangleq \{z_{1m}, \ldots, z_{Pm}\}^T$ is the mth input vector, P is the number of input variables, and y_m^d is the mth output of the function to be approximated. Let y be the output of the FLS approximator, $\mathcal{F}$, i.e., for $m = 1, \ldots, M$,

$$y(\mathbf{z}_m) \triangleq \mathcal{F}(\mathbf{z}_m) . \tag{5.16}$$

Let ϵ be the approximation error,

$$\epsilon(\mathbf{z}_m) \triangleq y(\mathbf{z}_m) - y_m^d . \tag{5.17}$$

Assume the approximation error, ϵ, to be a random variable normally distributed with mean, η_ϵ, and variance, σ_ϵ^2. Our objective is to determine the confidence limits, E^L and E^U, of the confidence interval, (E^L, E^U), for given confidence coefficient, γ, such that the probability of $\eta_\epsilon \in (E^L, E^U)$ is γ, i.e.,

$$P(E^L < \eta_\epsilon < E^U) = \gamma . \tag{5.18}$$

The sample variance of the random variable, ϵ, is

$$s_M^2 \triangleq \frac{1}{M-1} \sum_{m=1}^{M} \epsilon_m^2 . \tag{5.19}$$

where $\epsilon_m \triangleq \epsilon(\mathbf{z}_m)$. This is an unbiased estimate of the variance σ_ϵ^2, and it tends to σ_ϵ^2 as $M \to \infty$ [124, Chap.9]. Therefore, for large M,

$$s_M \approx \sigma_\epsilon . \tag{5.20}$$

The sample mean, $\bar{\epsilon}_M$, of the random variable, ϵ, is

$$\bar{\epsilon}_M \triangleq \frac{1}{M} \sum_{m=1}^{M} \epsilon_m . \tag{5.21}$$

For normally distributed random variable, ϵ, the confidence limits for its expected value, η_e, can be obtained as [124, Chap.9]

$$\begin{cases} E^U &= \bar{e}_M + t_{1-\frac{\delta}{2}} \frac{s_M}{\sqrt{M}} \\ E^L &= \bar{e}_M - t_{1-\frac{\delta}{2}} \frac{s_M}{\sqrt{M}} \end{cases} . \tag{5.22}$$

That is

$$P(\bar{e}_M - t_{1-\frac{\delta}{2}} \frac{s_M}{\sqrt{M}} < \eta_e < \bar{e}_M + t_{1-\frac{\delta}{2}} \frac{s_M}{\sqrt{M}}) = \gamma . \tag{5.23}$$

where $\delta \triangleq 1 - \gamma$, and $t_{1-\frac{\delta}{2}}$ is the $1 - \frac{\delta}{2}$ percentile[1] of a Student-t distribution with $M - 1$ degrees of freedom, which can be obtained from tables such as table 9-2 in [124].

5.2.3 Confidence Interval for the Mean of Prediction Errors of Individual IF-THEN Rules

In the following, for a given overall FLS output error, ϵ, the confidence intervals for the means of the prediction errors of individual rules are determined.

With a FLS using the CM defuzzifier, eq.(4.17), the output of the FLS is

$$y = \frac{\sum_{i=1}^{I} \mu_{Y^i}(\overline{y}^i) \cdot \overline{y}^i}{\sum_{i=1}^{I} \mu_{Y^i}(\overline{y}^i)} . \tag{5.24}$$

where Y^i, given in eq.(4.12), are the components of output fuzzy set induced by the ith rule, $\overline{y}^i$ is the center of a primary fuzzy set, B^i, in the output universe of discourse, and I is the size of the fuzzy rule base. Let

$$\theta_i \triangleq \frac{\mu_{Y^i}(\overline{y}^i)}{\sum_{i=1}^{I} \mu_{Y^i}(\overline{y}^i)} . \tag{5.25}$$

The output of the FLS, eq.(5.24), can be rewritten as

$$y = \sum_{i=1}^{I} \theta_i \overline{y}^i . \tag{5.26}$$

[1] The $1 - \frac{\delta}{2}$ percentile of a random variable, t, is the smallest number, $t_{1-\frac{\delta}{2}}$, such that $1 - \frac{\delta}{2} = P\{t \leq t_{1-\frac{\delta}{2}}\}$.

which is the weighted sum of the centers of the primary fuzzy sets in the output universe of discourse.

Assume, for the moment, that the total FLS output error, ϵ, can be distributed unambiguously among individual rules. Formally,

$$e \triangleq \sum_{i=1}^{I} \epsilon^i \ , \tag{5.27}$$

where ϵ^i is the fraction of ϵ contributed by the ith rule, and can be unambiguously determined given the total output error, ϵ.

One way to distribute the output error is to assume that the contribution of the ith rule to the total output error is proportional to the weighting coefficient, θ_i, i.e.,

$$\epsilon^i \triangleq \theta_i \cdot \epsilon \ . \tag{5.28}$$

For $i = 1, \ldots, I$, assume the ϵ^i to be random variables normally distributed with means, η_ϵ^i, and variances, σ_ϵ^i. Now, our objective is to find the confidence limits, $(E^L)^i$ and $(E^U)^i$, of the confidence interval, $((E^L)^i, (E^U)^i)$, for given confidence coefficients, γ^i, such that the probability of $\eta_\epsilon^i \in ((E^L)^i, (E^U)^i)$ is γ^i, i.e., for $i = 1, \ldots, I$,

$$P((E^L)^i < \eta_\epsilon^i < (E^U)^i) = \gamma^i \ . \tag{5.29}$$

The sample variances of the random variables, ϵ^i, are

$$(s_M^i)^2 \triangleq \frac{1}{M-1} \sum_{m=1}^{M} (e_m^i)^2 \ . \tag{5.30}$$

where $e_m^i \triangleq \epsilon^i(\mathbf{z}_m)$. The sample means, $\overline{e}_M^i$, of the ϵ^i are

$$\overline{e}_M^i \triangleq \frac{1}{M} \sum_{m=1}^{M} e_m^i \ . \tag{5.31}$$

The confidence limits for normally distributed variables, ϵ^i, are obtained as

$$\begin{cases} (E^U)^i &= \overline{\epsilon}_M^i + t_{1-\frac{\alpha^i}{2}} \frac{s_M^i}{\sqrt{M}} \ . \\ (E^L)^i &= \overline{\epsilon}_M^i - t_{1-\frac{\alpha^i}{2}} \frac{s_M^i}{\sqrt{M}} \end{cases} \tag{5.32}$$

That is

$$P(\bar{e}^i_M - t_{1-\frac{\delta^i}{2}}\frac{s^i_M}{\sqrt{M}} < \eta^i_e < \bar{e}^i_M + t_{1-\frac{\delta^i}{2}}\frac{s^i_M}{\sqrt{M}}) = \gamma^i \ . \tag{5.33}$$

where $\delta^i \triangleq 1-\gamma^i$, and $t_{1-\frac{\delta^i}{2}}$ is the $1-\frac{\delta^i}{2}$ percentile of a Student-t distribution with $M-1$ degrees of freedom.

5.2.4 Numerical Examples

Consider a discrete nonlinear system

$$f(z) = 3\sin(\pi z(k)) + \sin(3\pi z(k)) + v(k) \ . \tag{5.34}$$

where v is Gaussian noise with mean, $\eta_v = 0$, and standard deviation, $\sigma_v = 0.05$. z is a random variable uniformly distributed in $(-1, 1)$. $z(k)$ and $v(k)$ represent sampled values of random variables z and v, respectively, at time step k.

A FLS in the form of eq.(5.24) is used to approximate $f(z)$, whose structure is as shown in fig.5.4 with a single input and single output. Fifteen primary fuzzy sets.

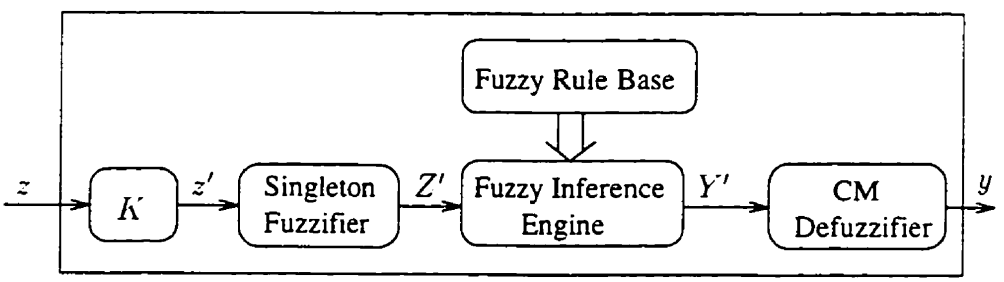


Figure 5.4: Structure of FLS Approximator

A_j, $j = 1, \ldots, 15$, are defined in the input universe of discourse, $\mathcal{Z} \triangleq (-1, 1)$, and Gaussian type membership functions are used for all the primary fuzzy sets, i.e.,

$$\mu_{A_i}(z') \triangleq \begin{cases} 1, & \text{if } z' < \bar{z}_1 \ . \\ \exp[-\frac{1}{2}(\frac{z'-\bar{z}_1}{\sigma_i})^2] \ . & \text{if } z' \geq \bar{z}_1 \ . \end{cases}$$

$$\mu_{A_j}(z') \triangleq \exp[-\frac{1}{2}(\frac{z'-\bar{z}_j}{\sigma_j})^2] \ , \ j = 2, 3, \ldots, 14, \tag{5.35}$$

$$\mu_{A_{15}}(z') \triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{z'-\bar{z}_{15}}{\sigma_{15}})^2] \ , & \text{if } z' < \bar{z}_{15} \ . \\ 1, & \text{if } z' \geq \bar{z}_{15} \ . \end{cases}$$

where $z' = Kz$, and $K \triangleq 7$ is a scaling parameter. Initially, the parameters, $\bar{z}_j$ and σ_j, are assigned the following values.

$$\begin{cases} \{z_1, z_2, \ldots, z_8, \ldots, z_{14}, z_{15}\}^T \triangleq \{-7, -6, \ldots, 0, \ldots, 6, 7\}^T \ , \\ \sigma_1 = \sigma_2 = \cdots = \sigma_{15} \triangleq 0.5 \ . \end{cases} \tag{5.36}$$

Fifteen primary fuzzy sets, B_l, $l = 1, \ldots, 15$, in output universe of discourse, $\mathcal{Y} \subset R$, are used and $\bar{y}_l$, $l = 1, \ldots, 15$, are the centers of the B_l, which are initially assigned random values uniformly distributed in $(-1, 1)$.

The output of this FLS is expressed as

$$y = \frac{\sum_{i=1}^{15} \mu_{A^i}(z') \cdot \bar{y}^i}{\sum_{i=1}^{15} \mu_{A^i}(z')} \ . \tag{5.37}$$

where $A^i \in \{A_j \mid A_j \in \mathcal{Z}, \ j = 1, \ldots, 15\}$, and $\bar{y}^i \in \{\bar{y}_l \mid \bar{y}_l \in \mathcal{Y}, \ l = 1, \ldots, 15\}$. Let

$$\theta_i \triangleq \frac{\mu_{A^i}(z')}{\sum_{i=1}^{15} \mu_{A^i}(z')} \ . \tag{5.38}$$

Then

$$y = \sum_{i=1}^{15} \theta_i \bar{y}^i \ . \tag{5.39}$$

A Confidence Interval for the Mean of FLS Approximation Errors

For purposes of comparison we use two training approaches to train the FLS: the back-propagation approach (BP), and a combined least squares estimation and back-propagation approach (LSE-BP). The BP approach uses the back-propagation algorithm to train parameters, $\bar{z}_j$, σ_j, and $\bar{y}_l$, $j, l = 1, \ldots, 15$. The LSE-BP approach uses BP training to train parameters $\bar{z}_j$, σ_j and $\bar{y}_l$ when LSE training is not applicable, and it uses LSE training to train $\bar{y}_l$ whenever LSE training is applicable.

The two training approaches are presented in detail in chapter 7, and here, we concentrate only on the statistical confidence measures of FLS performance.

Figure 5.5(a–b) shows the outputs of the FLS trained by LSE-BP and BP, respectively. In both BP and LSE-BP training, the learning gains for parameters $\bar{x}_j$, σ_j and $\bar{y}_l$ are all set to 0.1, i.e., for $j, l = 1, \ldots, 15$,

$$\alpha_{\bar{x}_j} = \alpha_{\sigma_j} = \alpha_{\bar{y}_l} \triangleq 0.1 \ . \tag{5.40}$$

In both cases, training terminates at $k = 37$. It is observed that the LSE-BP trained FLS has superior performance.

Let the approximation error of the FLS be ϵ, i.e.,

$$\epsilon(z) \triangleq y(z) - f(z) \ . \tag{5.41}$$

which is assumed normally distributed with mean, η_ϵ, and variance, σ_ϵ^2. Given a confidence coefficient,

$$\gamma \triangleq 0.9 \ . \tag{5.42}$$

then $\delta = 0.1$, and $t_{1-\frac{\delta}{2}} = t_{0.95}$. $t_{0.95}$ can be obtained from tables 9-1 and 9-2 in [124], and therefore the confidence limits for the means of FLS approximation errors, E^U and E^L, can be obtained with eq.(5.22). Figure 5.5(c) shows the 0.9 confidence intervals of η_e, for the FLS trained by LSE-BP and BP, respectively.

In fig.5.5(c), the bias of the confidence interval from horizontal axis represents the sample mean of FLS output error, and the width of the confidence interval is proportional to the estimated standard deviation of the output error. For both LSE-BP and BP trained FLS, the confidence intervals narrow down very quickly as training continues, which indicates the effectiveness of both training approaches. However, after the conclusion of training, the confidence interval of the LSE-BP trained FLS continues converging, and at a certain point has smaller width and smaller bias from the horizontal axis than that of the BP trained FLS, which indicates superior

performance of the LSE-BP trained FLS. This coincides with the observation made earlier on figs.5.5(a–b).

B Confidence Interval for the Mean of Prediction Errors of Individual Rules

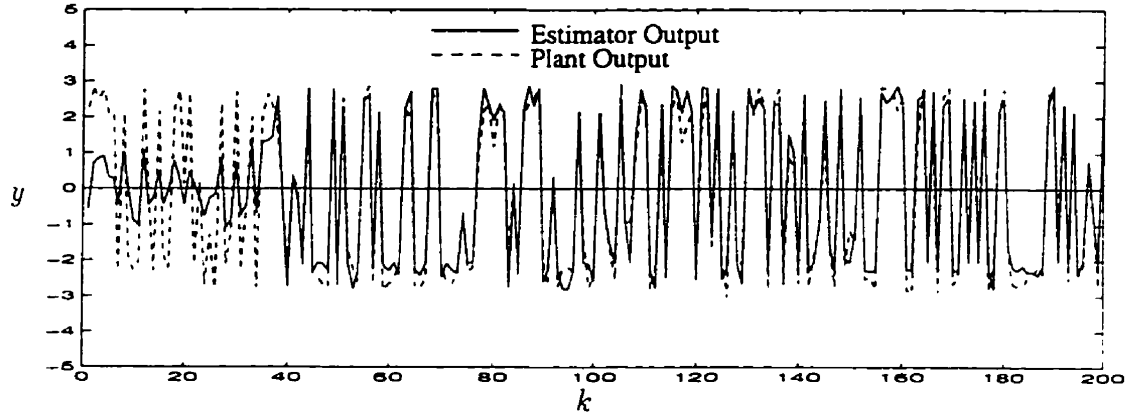
Assume the contribution of the ith rule to the total output error is proportional to the weighting coefficient, θ_i, i.e.,

$$\epsilon^i \triangleq \theta_i \cdot \epsilon \ , \tag{5.43}$$

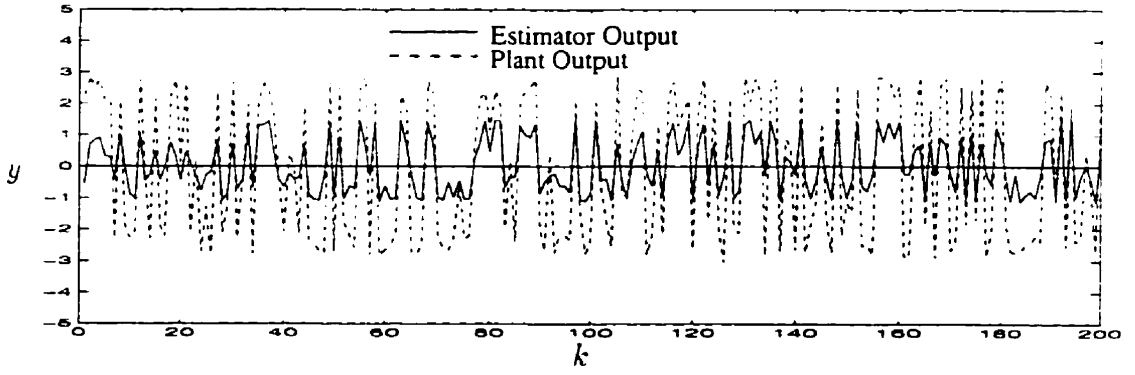
whose values are assumed normally distributed with mean, η_ϵ^i, and variance, $(\sigma_\epsilon^i)^2$.

Again, set confidence coefficients, γ^i, $i = 1, \ldots, 15$ at 0.9, so that $t_{1-\frac{\alpha^i}{2}} = t_{0.95}$. The confidence limits for the expected values of ϵ^i, $(E^U)^i$ and $(E^L)^i$, can be obtained with eq.(5.32). Considering only the FLS trained with the LSE-BP approach, figure 5.6 shows the confidence intervals for the odd numbered rules with respect to time step k. Figure 5.7 shows the confidence intervals of all the rules at time steps, k=50, 100, and 200.

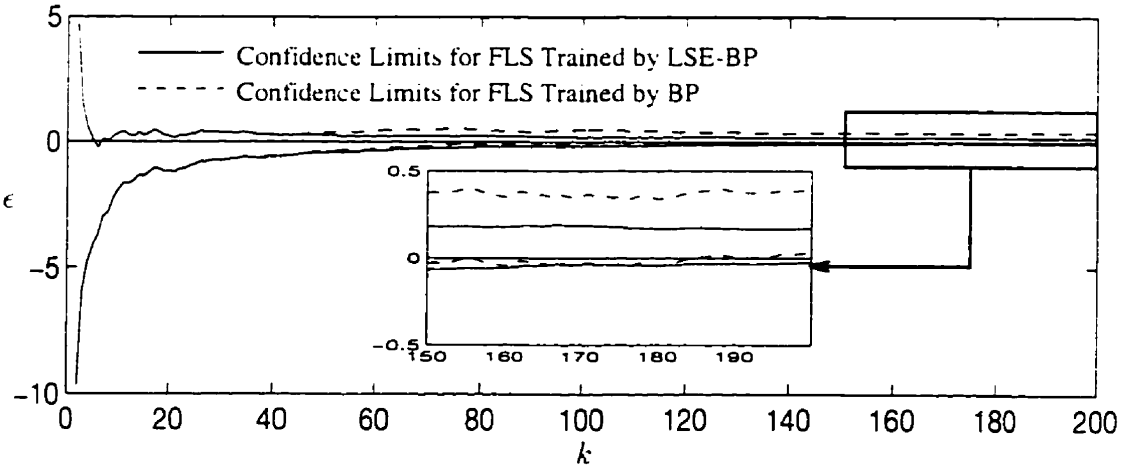
The error distribution approach used here is subjective, and how best to understand and make use of the confidence intervals of individual rules is yet to be investigated. We leave this as future work.



(a) Output of FLS Trained with LSE-BP Approach

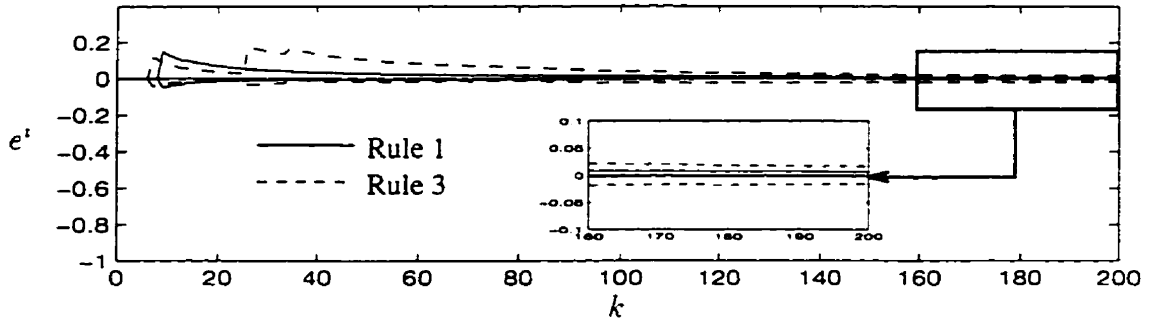


(b) Output of FLS Trained with BP Approach

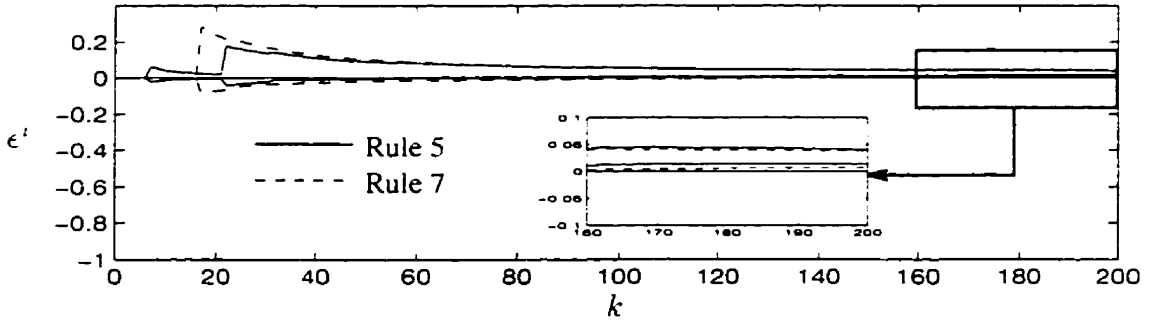


(c) Confidence Intervals for Expected Values of Approaximation Errors

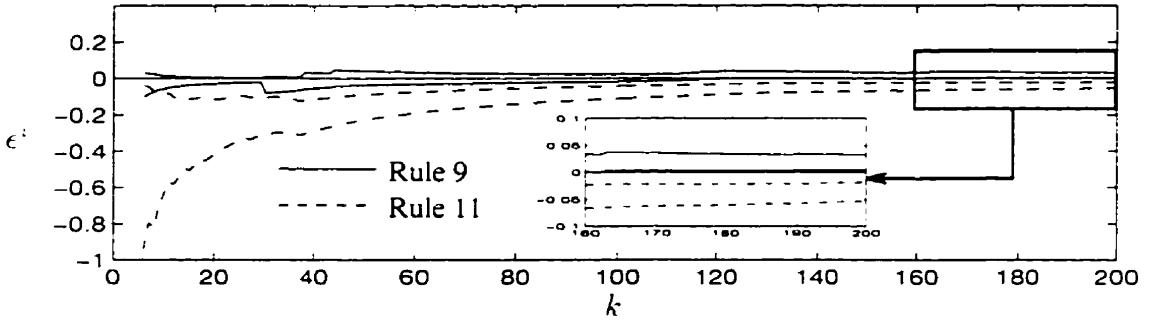
Figure 5.5: FLS Outputs and Confidence Intervals for Expected Values of Approximation Errors



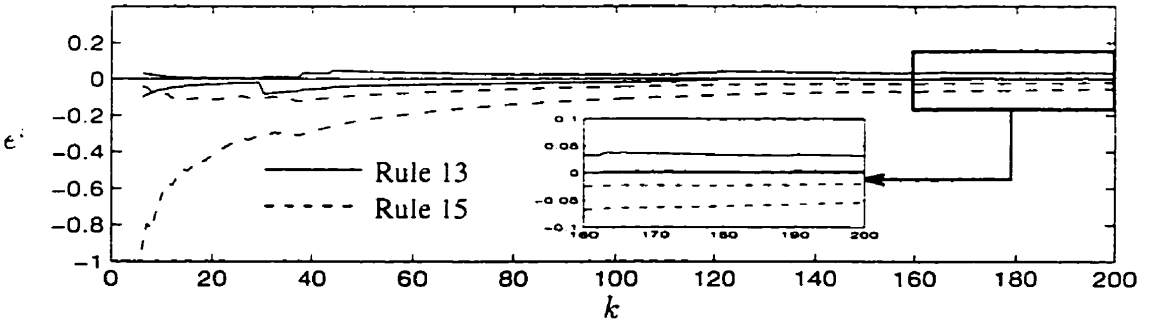
(a) Confidence Intervals for Rules 1 and 3



(b) Confidence Intervals for Rules 5 and 7



(c) Confidence Intervals for Rules 9 and 11



(d) Confidence Intervals for Rules 13 and 15

Figure 5.6: Confidence Intervals for Expected Values of Prediction Errors of Individual Rules

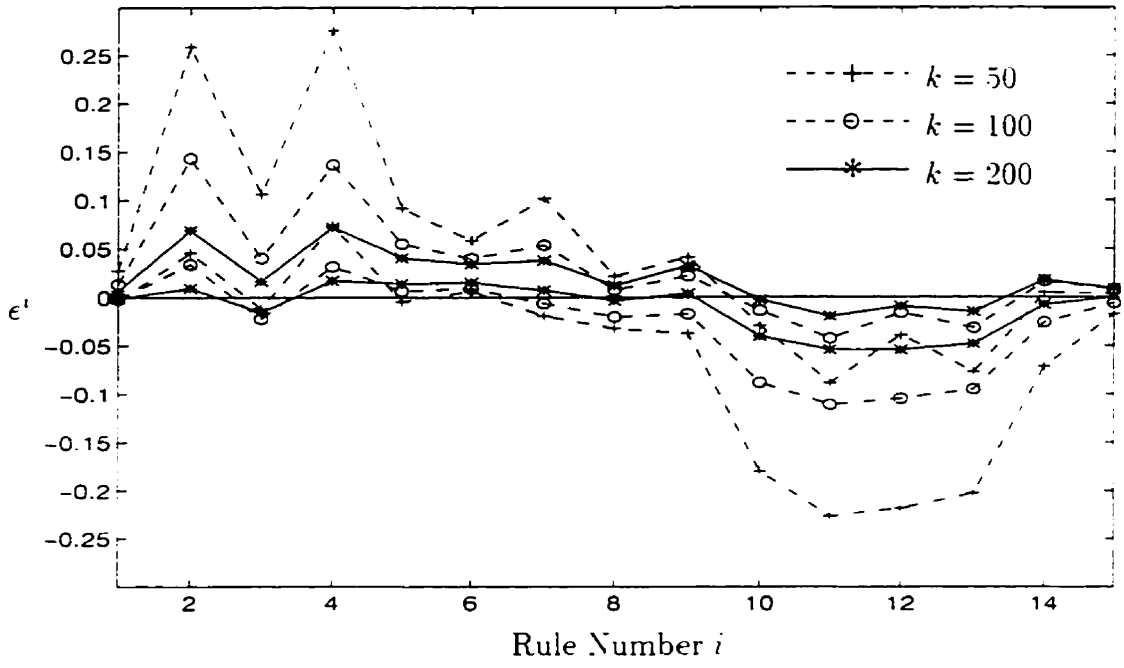


Figure 5.7: Confidence Intervals for Individual Rules at k=50, 100, and 200

5.3 On Computational Issues of Fuzzy Logic Systems

5.3.1 Introduction

Consider the FLS-II of section 4.6.2, which is a FLS with singleton fuzzifier, CM defuzzifier, sup-product compositional rule of inference, algebraic product operation for fuzzy implications and t-norms. If further Gaussian type membership functions for primary fuzzy sets are used, the output of this FLS is given by eq.(4.42) as

$$y \triangleq \frac{\sum_{i=1}^{I} \overline{y}^i \prod_{p=1}^{P} \exp[-(\frac{z_p - \overline{z}_p^i}{\sigma_p^i})^2]}{\sum_{i=1}^{I} \prod_{p=1}^{P} \exp[-(\frac{z_p - \overline{z}_p^i}{\sigma_p^i})^2]} \ . \tag{5.44}$$

This FLS will be extensively used later in this work. In the following, we present an approach for simplifying the calculation of eq.(5.44) and increasing speed of computation, which is much desired for on-line FLS control algorithms where calculation speed is often critical for success. To address the concern of the error resulting from this simplification, the bounds of the errors will also be specified and tabulated for certain situations.

5.3.2 Approximation in FLS Computation

Let

$$\theta_i \triangleq \frac{\prod_{p=1}^{P} \exp[-(\frac{z_p - \overline{z}_p^i}{\sigma_p^i})^2]}{\sum_{i=1}^{I} \prod_{p=1}^{P} \exp[-(\frac{z_p - \overline{z}_p^i}{\sigma_p^i})^2]} \ , \tag{5.45}$$

where $z_p \in \mathcal{Z}_p$, $p = 1, \ldots, P$, are input variables, $\overline{z}_p^i$ are the centers of the primary fuzzy sets in the universes of discourse $\mathcal{Z}_p$ and appear in the ith rule. Rewrite eq.(5.44) as

$$y = \sum_{i=1}^{I} \theta_i \overline{y}^i \ . \tag{5.46}$$

where θ_i is a coefficient determined by the IF part of the ith rule, and $\overline{y}^i$ is the center of the primary fuzzy set in the output universe of discourse used in the THEN part of the ith rule. The term, $(\theta_i \overline{y}^i)$, represents the contribution of the ith rule to the total output.

Consider a fuzzy partition for a given universe of discourse, $\mathcal{Z}_p$, as illustrated in fig.5.8. For $p = 1, \ldots, P$, and $j_p = 1, \ldots, J_p$, $\overline{z}_{pj_p} \in \mathcal{Z}_p$ represent the centers of the

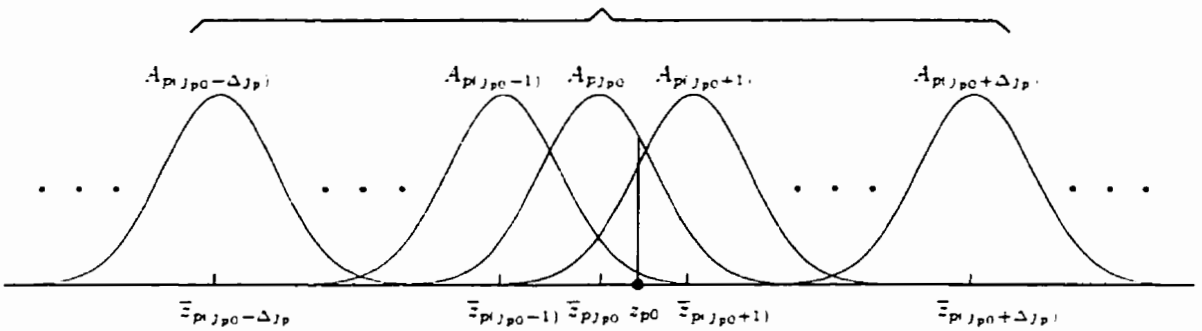


Figure 5.8: A Fuzzy Partition

primary fuzzy sets, $A_{pj_p} \in \mathcal{Z}_p$, and $\overline{z}_p^i \in \{\overline{z}_{pj_p} \mid j_p = 1, \ldots, J_p\}$.

For a given crisp input $z_{p0} \in \mathcal{Z}_p$, the grade of membership of z_{p0} in A_{pj_p} is

$$\mu_{A_{pj_p}}(z_{p0}) = \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2] \,. \tag{5.47}$$

which decreases exponentially with increasing separation between z_{p0} and $\overline{z}_{pj_p}$. Intuitively, for $p \in \{1, \ldots, P\}$ and $j_p \in \{1, \ldots, J_p\}$, if $| z_{p0} - \overline{z}_{pj_p} |$ is *large enough* such that $\mu_{A_{pj_p}}(z_{p0})$ is *small enough*, then the corresponding coefficients, θ_i, $i \in \{1, \ldots, I\}$, which contain $\mu_{A_{pj_p}}(z_{p0})$ are sufficiently small that the effects of the relevant IF-THEN rules are negligible. If a relatively large number of fuzzy rules can be neglected for each input without causing significant error, the calculation speed can increase significantly with acceptable output accuracy.

Developing this notion further, we now assume that only those primary fuzzy sets in the neighbourhood of the crisp input, z_{p0}, say from $A_{p(j_{p0}-\Delta j_p)}$ to $A_{p(j_{p0}+\Delta j_p)}$, are required in calculating the FLS output, as shown in fig.5.8, and the resulting

approximation error of the overall FLS output from neglecting the rest of the primary fuzzy sets, i.e., those outside the indicated neighbourhood, is negligible.

From observing fig.5.8, it is clear that the range of the indicated neighbourhood, Δ_{j_p}, and the shape of the membership functions, σ_{pj_p}, determine the FLS output accuracy (or approximation error).

Next, we investigate the bounds of the approximation errors for different ranges of neighbourhood and different shapes of membership functions, (i.e., for different values of Δ_{j_p} and σ_{pj_p}), which provide some guidance in determining which rules can be neglected in calculations while still maintaining acceptable output accuracy.

5.3.3 Bounds of Approximation Errors

An expression for the bounds of the approximation errors for general fuzzy logic systems is difficult to derive. We confine our derivation to a two-input single-output FLS, i.e.,

$$P \triangleq 2 \ . \tag{5.48}$$

because, first, this type of systems has been widely used in various application (e.g., with inputs being "error" and "change of error"), and second, in the case of more than two inputs the analysis procedure here is still applicable (except the process is much more tedious).

Another restriction on our scope is that it is assumed for all the universes of discourse that the centers of the membership functions of the primary fuzzy sets are evenly spaced with unit distances, i.e., for $p \in \{1,2\}$ and $j_p \in \{2,\ldots,J_p\}$,

$$| \bar{z}_{pj_p} - \bar{z}_{p(j_p-1)} | \triangleq 1 \ . \tag{5.49}$$

Again, this is for simplicity of derivation. The derived results are directly applicable to most situations in this work, for which the centers of the primary fuzzy sets in input space are usually evenly spaced. For situations other than these, although the

numerical results may not be directly applicable, they can nevertheless reveal the level of the approximation errors. At the very least, the analysis procedure here is still valid and may be applied in specific situations to rederive the numerical results.

Define primary fuzzy sets, $\{A_{1j_1} \mid j_1 = 1, \ldots, J_1\}$ and $\{A_{2j_2} \mid j_2 = 1, \ldots, J_2\}$, in input universes of discourse, $\mathcal{Z}_1$ and $\mathcal{Z}_2$, respectively. The IF-THEN rules have the form

$$\text{IF } z_1 \text{ is } A_{1j_1} \text{ and } z_2 \text{ is } A_{2j_2} \text{ THEN } y \text{ is } B_{j_1 j_2} .$$

where $B_{j_1 j_2} \in \mathcal{Y}$ is an output primary fuzzy set, whose center is denoted as $\bar{y}_{j_1 j_2}$, and $\mathcal{Y}$ is the output universe of discourse. This fuzzy rule base is illustrated in table 5.2, which is composed of $J_1 \times J_2$ rules.

Table 5.2: Fuzzy Rule Base

$\mathcal{Y}$ / $\mathcal{Z}_2$ / $\mathcal{Z}_1$	A_{21}	$\cdots$	$A_{2(j_{20}-\Delta_{j_2})}$	$\cdots$	$A_{2j_{20}}$	$\cdots$	$A_{2(j_{20}+\Delta_{j_2})}$	$\cdots$	A_{2J_2}
A_{11}	B_{11}	$\cdots$	$B_{1(j_{20}-\Delta_{j_2})}$	$\cdots$	$B_{1j_{20}}$	$\cdots$	$B_{1(j_{20}+\Delta_{j_2})}$	$\cdots$	B_{1J_2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A_{1(j_{10}-\Delta_{j_1})}$									
$\vdots$									
$A_{1j_{10}}$									
$\vdots$									
$A_{1(j_{10}+\Delta_{j_1})}$									
$\vdots$									
A_{1J_1}	B_{J_11}	$\cdots$	$B_{J_1(j_{20}-\Delta_{j_2})}$	$\cdots$	$B_{J_1j_{20}}$	$\cdots$	$B_{J_1(j_{20}+\Delta_{j_2})}$	$\cdots$	$B_{J_1J_2}$

For given crisp input, $\{z_{10}, z_{20}\}$ we assume $\{\bar{z}_{1j_{10}}, \bar{z}_{2j_{20}}\}$ to be the closest centers

of membership functions of primary fuzzy sets, $\{A_{1j_{10}}, A_{2j_{20}}\}$. We also assume that only the primary fuzzy sets between $A_{p(j_{p0}-\Delta j_p)}$ and $A_{p(j_{p0}+\Delta j_p)}$, $p = 1, 2$, are used in calculations for this input. The relevant IF-THEN rules correspond to the shaded area in table 5.2, and other rules are then neglected. It should be noted that the subscripts, j_{p0} and Δj_p, must satisfy the following conditions, for $p = 1, 2$,

$$\begin{cases} j_{p0} - \Delta j_p \geq 1 , \\ j_{p0} + \Delta j_p \leq J_p . \end{cases} \tag{5.50}$$

We now rewrite eq.(5.44) as

$$y = \frac{\sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \overline{y}_{j_1 j_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2]}{\sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2]} . \tag{5.51}$$

and make the following definitions:

$$S_n \triangleq \sum_{j_1=j_{10}-\Delta j_1}^{j_{10}+\Delta j_1} \sum_{j_2=j_{20}-\Delta j_2}^{j_{20}+\Delta j_2} \overline{y}_{j_1 j_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2] , \tag{5.52}$$

$$S_d \triangleq \sum_{j_1=j_{10}-\Delta j_1}^{j_{10}+\Delta j_1} \sum_{j_2=j_{20}-\Delta j_2}^{j_{20}+\Delta j_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2] . \tag{5.53}$$

$$\begin{aligned} \Delta_n \triangleq & \sum_{j_1=1}^{j_{10}-\Delta j_1-1} \sum_{j_2=1}^{J_2} \overline{y}_{j_1 j_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2] \\ & + \sum_{j_1=j_{10}-\Delta j_1}^{j_{10}+\Delta j_1} \{ \sum_{j_2=1}^{j_{20}-\Delta j_2-1} \overline{y}_{j_1 j_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2] \\ & \qquad + \sum_{j_2=j_{20}+\Delta j_2+1}^{J_2} \overline{y}_{j_1 j_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2] \} \\ & + \sum_{j_1=j_{10}+\Delta j_1+1}^{J_1} \sum_{j_2=1}^{J_2} \overline{y}_{j_1 j_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2] . \end{aligned} \tag{5.54}$$

$$\begin{aligned} \Delta_d \triangleq & \sum_{j_1=1}^{j_{10}-\Delta j_1-1} \sum_{j_2=1}^{J_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2] \\ & + \sum_{j_1=j_{10}-\Delta j_1}^{j_{10}+\Delta j_1} \{ \sum_{j_2=1}^{j_{20}-\Delta j_2-1} \prod_{p=1}^{2} \exp[-(\frac{z_{p0} - \overline{z}_{pj_p}}{\sigma_{pj_p}})^2] \end{aligned}$$

$$+ \sum_{j_2=j_{20}+\Delta_{j_2}+1}^{J_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0}-\overline{z}_{pj_p}}{\sigma_{pj_p}})^2]\}$$

$$+ \sum_{j_1=j_{10}+\Delta_{j_1}+1}^{J_1} \sum_{j_2=1}^{J_2} \prod_{p=1}^{2} \exp[-(\frac{z_{p0}-\overline{z}_{pj_p}}{\sigma_{pj_p}})^2] \ . \tag{5.55}$$

Using these symbols in eq.(5.51) results in

$$y = \frac{S_n + \Delta_n}{S_d + \Delta_d} \ . \tag{5.56}$$

From table 5.2 it is clear that S_n and S_d are relevant to the rules in the shaded area, while Δ_n and Δ_d are relevant to the rules outside the shaded area and represent the residual values of the numerator and denominator of eq.(5.51), respectively, which resulted from the approximations.

Let

$$y' \triangleq \frac{S_n}{S_d} \ . \tag{5.57}$$

which is the output approximated by using only those rules in the shaded area in table 5.2. The approximation error is

$$\Delta y \triangleq y' - y \ . \tag{5.58}$$

Denote, for $p = 1, 2$,

$$\begin{cases} y_{\max} \triangleq \sup_{y \in \mathcal{Y}}\{|\ y\ |\} \ , \\ \overline{y}_{\max} \triangleq \max\{|\ \overline{y}_{j_1 j_2}\ |\ |\ \overline{y}_{j_1 j_2} \in \mathcal{Y},\ j_1 = 1, \ldots, J_1,\ j_2 = 1, \ldots, J_2\} \ . \\ \sigma_{p\max} \triangleq \max\{\sigma_{pj_p} \mid j_p = 1, \ldots, J_p\} \ . \\ \sigma_{p\min} \triangleq \min\{\sigma_{pj_p} \mid j_p = 1, \ldots, J_p\} \ . \end{cases} \tag{5.59}$$

We then have the following theorem.

Theorem 5.1 *Consider the FLS of eq.(5.44) subject to the restrictions of eqs.(5.48-5.49). For a given crisp input* $\{z_{10},\ z_{20}\}$, *if only the rules in the shaded area in table*

5.2 are used in calculating the corresponding output, y, the absolute value of the approximation error, Δy, is bounded by $2y_{\max}\frac{\Delta_{d\max}}{S_{d\min}}$, i.e.,

$$\frac{|\Delta y|}{y_{\max}} \leq 2\frac{\Delta_{d\max}}{S_{d\min}} \ . \tag{5.60}$$

where

$$\begin{aligned}
\Delta_{d\max} \triangleq\ & \{\sum_{j_2=1}^{j_{20}-1}\exp[-(\frac{\bar{z}_{2j_{20}}-\bar{z}_{2j_2}-0.5}{\sigma_{2\max}})^2]+1 \\
& +\sum_{j_2=j_{20}+1}^{J_2}\exp[-(\frac{\bar{z}_{2j_2}-\bar{z}_{2j_{20}}-0.5}{\sigma_{2\max}})^2]\} \\
& \times\{\sum_{j_1=1}^{j_{10}-\Delta_{j_1}-1}\exp[-(\frac{\bar{z}_{1j_{10}}-\bar{z}_{1j_1}-0.5}{\sigma_{1\max}})^2] \\
& +\sum_{j_1=j_{10}+\Delta_{j_1}+1}^{J_1}\exp[-(\frac{\bar{z}_{1j_1}-\bar{z}_{1j_{10}}-0.5}{\sigma_{1\max}})^2]\} \\
+\ & \{\sum_{j_1=j_{10}-\Delta_{j_1}}^{j_{10}-1}\exp[-(\frac{\bar{z}_{1j_{10}}-\bar{z}_{1j_1}-0.5}{\sigma_{1\max}})^2]+1 \\
& +\sum_{j_1=j_{10}+1}^{j_{10}+\Delta_{j_1}}\exp[-(\frac{\bar{z}_{1j_1}-\bar{z}_{1j_{10}}-0.5}{\sigma_{1\max}})^2]\} \\
& \times\{\sum_{j_2=1}^{j_{20}-\Delta_{j_2}-1}\exp[-(\frac{\bar{z}_{2j_{20}}-\bar{z}_{2j_2}-0.5}{\sigma_{2\max}})^2] \\
& +\sum_{j_2=j_{20}+\Delta_{j_2}+1}^{J_2}\exp[-(\frac{\bar{z}_{2j_2}-\bar{z}_{2j_{20}}-0.5}{\sigma_{2\max}})^2]\} \ .
\end{aligned} \tag{5.61}$$

$$\begin{aligned}
S_{d\min} \triangleq\ & \{\sum_{j_1=j_{10}-\Delta_{j_1}}^{j_{10}-1}\exp[-(\frac{\bar{z}_{1j_{10}}-\bar{z}_{1j_1}+0.5}{\sigma_{1\min}})^2]+\exp[-(\frac{0.5}{\sigma_{1\min}})^2] \\
& +\sum_{j_1=j_{10}+1}^{j_{10}+\Delta_{j_1}}\exp[-(\frac{\bar{z}_{1j_1}-\bar{z}_{1j_{10}}+0.5}{\sigma_{1\min}})^2]\} \\
& \times\{\sum_{j_2=j_{20}-\Delta_{j_2}}^{j_{20}-1}\exp[-(\frac{\bar{z}_{2j_{20}}-\bar{z}_{2j_2}+0.5}{\sigma_{2\min}})^2]+\exp[-(\frac{0.5}{\sigma_{2\min}})^2] \\
& +\sum_{j_2=j_{20}+1}^{j_{20}+\Delta_{j_2}}\exp[-(\frac{\bar{z}_{2j_2}-\bar{z}_{2j_{20}}+0.5}{\sigma_{2\min}})^2]\}
\end{aligned} \tag{5.62}$$

The proof of this theorem is presented in the next subsection (5.3.4).

For given parameters, $\sigma_{p\max}$, $\sigma_{p\min}$, and Δj_p, $p = 1, 2$, the bounds of the approximation error, $\frac{|\Delta y|}{y_{\max}}$, can be calculated with this theorem.

The special case in which

$$\sigma_{1\max} = \sigma_{1\min} = \sigma_{2\max} = \sigma_{2\min} \triangleq \sigma \ , \qquad \Delta j_1 = \Delta j_2 \triangleq \Delta j \ , \tag{5.63}$$

(provided here to allow the reader to get a feel for this), has bounds of error for different values of σ and Δj calculated and presented in table 5.3. Other cases may

Table 5.3: Bounds of Approximation Errors

$\frac{\lvert\Delta y\rvert}{y_{\max}}$ \ σ / Δj	0.3	0.35	0.40	0.45	0.50	0.55	0.65
1	3.23×10^{-8}	6.30×10^{-6}	2.02×10^{-4}	2.23×10^{-3}	1.26×10^{-2}	4.59×10^{-2}	2.59×10^{-1}
2	1.61×10^{-27}	4.15×10^{-20}	2.80×10^{-15}	5.89×10^{-12}	1.42×10^{-9}	8.30×10^{-8}	2.01×10^{-5}

also be calculated from theorem 5.1. Table 5.3 may also serve as a nominal case which illustrates the level of the approximation errors.

Remarks

- Theorem 5.1 presents bounds of approximation error for the situation of two input variables. In the case of more than two input variables, the analysis to obtain the bounds of the output error is the same, but more tedious.

- From table 5.3, it is seen that if $\Delta j = 1$, and $\sigma \leq 0.45$, the approximation error, $\frac{|\Delta y|}{y_{\max}}$, is no greater than 0.223%, which is satisfactory for many practical applications. In this situation, only three primary fuzzy sets in each universe of discourse need be used in FLS calculations for a given crisp input. From experience, we have also found that use of the three primary fuzzy sets closest to the crisp input speeds up calculation without significantly sacrificing accuracy. Observation of fig.5.8 also suggests that the three primary fuzzy sets

closest to the crisp input contribute most significantly. Theorem 5.1 and table 5.3 justify our experience and intuition.

- In the case of two input variables, if seven primary fuzzy sets are defined in each universes of discourse, e.g., as in [91], then there are forty nine rules in the rule base. For any given crisp input, if three primary fuzzy sets in each of the input universes of discourse are used rather than seven, this corresponds to nine rules in the rule base, which only counts for 18.4% of the entire rule base. Therefore, calculation speed can be increased significantly, which is much desired for on-line control applications.

5.3.4 Proof of Theorem 5.1

From eq.(5.58),

$$| \Delta y | = | y' - y | = | \frac{S_n}{S_d} - \frac{S_n + \Delta_n}{S_d + \Delta_d} | = | \frac{\Delta_n}{S_d + \Delta_d} - y' \frac{\Delta_d}{S_d + \Delta_d} | \tag{5.64}$$

Since $S_i > 0$, $\Delta_d \geq 0$,

$$| \Delta y | \leq \frac{| \Delta_n - y' \Delta_d |}{S_i} \leq \frac{| \Delta_n |}{S_d} + | y' | \frac{\Delta_d}{S_d} \leq \frac{| \Delta_n |}{S_d} + y_{\max} \frac{\Delta_d}{S_d} . \tag{5.65}$$

From definitions of $\overline{y}_{\max}$ and $y_{\max}$, eq.(5.59), then,

$$\overline{y}_{\max} \leq y_{\max} . \tag{5.66}$$

Using eqs. (5.59) and (5.66) in eq.(5.54) yields

$$\begin{aligned} | \Delta_n | \leq \; & y_{\max} \sum_{j_2=1}^{J_2} \exp[-(\frac{z_{20} - \overline{z}_{2j_2}}{\sigma_{2j_2}})^2] \{ \sum_{j_1=1}^{j_{10}-\Delta_{j_1}-1} \exp[-(\frac{z_{10} - \overline{z}_{1j_1}}{\sigma_{1j_1}})^2] \\ & + \sum_{j_1=j_{10}+\Delta_{j_1}+1}^{J_1} \exp[-(\frac{z_{10} - \overline{z}_{1j_1}}{\sigma_{1j_1}})^2] \} \\ & + y_{\max} \sum_{j_1=j_{10}-\Delta_{j_1}}^{j_{10}+\Delta_{j_1}} \exp[-(\frac{z_{10} - \overline{z}_{1j_1}}{\sigma_{1j_1}})^2] \{ \sum_{j_2=1}^{j_{20}-\Delta_{j_2}-1} \exp[-(\frac{z_{20} - \overline{z}_{2j_2}}{\sigma_{2j_2}})^2] \\ & + \sum_{j_2=j_{20}+\Delta_{j_2}+1}^{J_2} \exp[-(\frac{z_{20} - \overline{z}_{2j_2}}{\sigma_{2j_2}})^2] \} . \end{aligned} \tag{5.67}$$

Since, for $p = 1, 2$, $\bar{z}_{pj_{p0}}$ are the closest to z_{p0} among the centers of the primary fuzzy sets.

$$| z_{p0} - \bar{z}_{pj_{p0}} | \leq 0.5 , \tag{5.68}$$

Or,

$$\bar{z}_{pj_{p0}} - 0.5 \leq z_{p0} \leq \bar{z}_{pj_{p0}} + 0.5 . \tag{5.69}$$

For $\bar{z}_{pj_p} < \bar{z}_{pj_{p0}}$, then

$$0 < \bar{z}_{pj_{p0}} - \bar{z}_{pj_p} - 0.5 \leq z_{p0} - \bar{z}_{pj_p} \leq \bar{z}_{pj_{p0}} - \bar{z}_{pj_p} + 0.5 . \tag{5.70}$$

Therefore,

$$\exp[-(\frac{\bar{z}_{pj_{p0}} - \bar{z}_{pj_p} + 0.5}{\sigma_{p\min}})^2] \leq \exp[-(\frac{z_{p0} - \bar{z}_{pj_p}}{\sigma_{pj_p}})^2] \leq \exp[-(\frac{\bar{z}_{pj_{p0}} - \bar{z}_{pj_p} - 0.5}{\sigma_{p\max}})^2]; \tag{5.71}$$

while for the case of $\bar{z}_{pj_p} = \bar{z}_{pj_{p0}}$,

$$\exp[-(\frac{0.5}{\sigma_{p\min}})^2] \leq \exp[-(\frac{z_{p0} - \bar{z}_{pj_p}}{\sigma_{pj_p}})^2] \leq 1 ; \tag{5.72}$$

and for $\bar{z}_{pj_p} > \bar{z}_{pj_{p0}}$,

$$0 < \bar{z}_{pj_p} - \bar{z}_{pj_{p0}} - 0.5 \leq -(z_{p0} - \bar{z}_{pj_p}) \leq \bar{z}_{pj_p} - \bar{z}_{pj_{p0}} + 0.5 , \tag{5.73}$$

so that,

$$\exp[-(\frac{\bar{z}_{pj_p} - \bar{z}_{pj_{p0}} + 0.5}{\sigma_{p\min}})^2] \leq \exp[-(\frac{z_{p0} - \bar{z}_{pj_p}}{\sigma_{pj_p}})^2] \leq \exp[-(\frac{\bar{z}_{pj_p} - \bar{z}_{pj_{p0}} - 0.5}{\sigma_{p\max}})^2]; \tag{5.74}$$

Using eqs.(5.71), (5.72) and (5.74) with eq.(5.67) yields

$$| \Delta_n | \leq y_{\max}\{\sum_{j_2=1}^{j_{20}-1} \exp[-(\frac{\bar{z}_{2j_{20}} - \bar{z}_{2j_2} - 0.5}{\sigma_{2\max}})^2] + 1$$

$$
\begin{aligned}
&+ \sum_{j_2=j_{20}+1}^{J_2} \exp[-(\frac{\bar{z}_{2j_2} - \bar{z}_{2j_{20}} - 0.5}{\sigma_{2\max}})^2]\} \\
&\times \{ \sum_{j_1=1}^{j_{10}-\Delta_{j_1}-1} \exp[-(\frac{\bar{z}_{1j_{10}} - \bar{z}_{1j_1} - 0.5}{\sigma_{1\max}})^2] \\
&+ \sum_{j_1=j_{10}+\Delta_{j_1}+1}^{J_1} \exp[-(\frac{\bar{z}_{1j_1} - \bar{z}_{1j_{10}} - 0.5}{\sigma_{1\max}})^2]\} \\
&+ y_{\max}\{ \sum_{j_1=j_{10}-\Delta_{j_1}}^{j_{10}-1} \exp[-(\frac{\bar{z}_{1j_{10}} - \bar{z}_{1j_1} - 0.5}{\sigma_{1\max}})^2] + 1 \\
&+ \sum_{j_1=j_{10}+1}^{j_{10}+\Delta_{j_1}} \exp[-(\frac{\bar{z}_{1j_1} - \bar{z}_{1j_{10}} - 0.5}{\sigma_{1\max}})^2]\} \\
&\cdot \{ \sum_{j_2=1}^{j_{20}-\Delta_{j_2}-1} \exp[-(\frac{\bar{z}_{2j_{20}} - \bar{z}_{2j_2} - 0.5}{\sigma_{2\max}})^2] \\
&+ \sum_{j_2=j_{20}+\Delta_{j_2}+1}^{J_2} \exp[-(\frac{\bar{z}_{2j_2} - \bar{z}_{2j_{20}} - 0.5}{\sigma_{2\max}})^2]\} . \qquad (5.75)
\end{aligned}
$$

Using eqs. (5.71), (5.72) and (5.74) with eq.(5.55) yields

$$
\begin{aligned}
\Delta_d \leq &\{ \sum_{j_2=1}^{j_{20}-1} \exp[-(\frac{\bar{z}_{2j_{20}} - \bar{z}_{2j_2} - 0.5}{\sigma_{2\max}})^2] + 1 \\
&+ \sum_{j_2=j_{20}+1}^{J_2} \exp[-(\frac{\bar{z}_{2j_2} - \bar{z}_{2j_{20}} - 0.5}{\sigma_{2\max}})^2]\} \\
&\times \{ \sum_{j_1=1}^{j_{10}-\Delta_{j_1}-1} \exp[-(\frac{\bar{z}_{1j_{10}} - \bar{z}_{1j_1} - 0.5}{\sigma_{1\max}})^2] \\
&+ \sum_{j_1=j_{10}+\Delta_{j_1}+1}^{J_1} \exp[-(\frac{\bar{z}_{1j_1} - \bar{z}_{1j_{10}} - 0.5}{\sigma_{1\max}})^2]\} \\
&+ \{ \sum_{j_1=j_{10}-\Delta_{j_1}}^{j_{10}-1} \exp[-(\frac{\bar{z}_{1j_{10}} - \bar{z}_{1j_1} - 0.5}{\sigma_{1\max}})^2] + 1 \\
&+ \sum_{j_1=j_{10}+1}^{j_{10}+\Delta_{j_1}} \exp[-(\frac{\bar{z}_{1j_1} - \bar{z}_{1j_{10}} - 0.5}{\sigma_{1\max}})^2]\} \\
&\times \{ \sum_{j_2=1}^{j_{20}-\Delta_{j_2}-1} \exp[-(\frac{\bar{z}_{2j_{20}} - \bar{z}_{2j_2} - 0.5}{\sigma_{2\max}})^2] \\
&+ \sum_{j_2=j_{20}+\Delta_{j_2}+1}^{J_2} \exp[-(\frac{\bar{z}_{2j_2} - \bar{z}_{2j_{20}} - 0.5}{\sigma_{2\max}})^2]\} . \qquad (5.76)
\end{aligned}
$$

Using eqs. (5.71), (5.72) and (5.74) with eq.(5.53) results in

$$
\begin{aligned}
S_d \;\geq\; & \Big\{ \sum_{j_1=j_{10}-\Delta_{j1}}^{j_{10}-1} \exp\Big[-\Big(\frac{\bar{z}_{1j_{10}} - \bar{z}_{1j_1} + 0.5}{\sigma_{1\min}}\Big)^2\Big] + \exp\Big[-\Big(\frac{0.5}{\sigma_{1\min}}\Big)^2\Big] \\
& + \sum_{j_1=j_{10}+1}^{j_{10}+\Delta_{j1}} \exp\Big[-\Big(\frac{\bar{z}_{1j_1} - \bar{z}_{1j_{10}} + 0.5}{\sigma_{1\min}}\Big)^2\Big]\Big\} \\
& \times \Big\{ \sum_{j_2=j_{20}-\Delta_{j2}}^{j_{20}-1} \exp\Big[-\Big(\frac{\bar{z}_{2j_{20}} - \bar{z}_{2j_2} + 0.5}{\sigma_{2\min}}\Big)^2\Big] + \exp\Big[-\Big(\frac{0.5}{\sigma_{2\min}}\Big)^2\Big] \\
& + \sum_{j_2=j_{20}+1}^{j_{20}+\Delta_{j2}} \exp\Big[-\Big(\frac{\bar{z}_{2j_2} - \bar{z}_{2j_{20}} + 0.5}{\sigma_{2\min}}\Big)^2\Big]\Big\} \; . \qquad (5.77)
\end{aligned}
$$

Now eqs. (5.75) and (5.76) in the light of (5.61) respectively give

$$
\begin{cases} |\,\Delta_n\,| \;\leq\; y_{\max}\Delta_{d\max} \; . \\ \Delta_d \;\leq\; \Delta_{d\max} \; . \end{cases} \qquad (5.78)
$$

and eq.(5.62) with eq.(5.77) shows that

$$
S_d \geq S_{d\min} \; . \qquad (5.79)
$$

Finally eqs.(5.78) and (5.79) with eq.(5.65) yields

$$
|\,\Delta y\,| \leq 2y_{\max}\frac{\Delta_{d\max}}{S_{d\min}} \; . \qquad (5.80)
$$

Or,

$$
\frac{|\,\Delta y\,|}{y_{\max}} \leq 2\frac{\Delta_{d\max}}{S_{d\min}} \; . \qquad (5.81)
$$

This completes the proof.

Chapter 6

Fuzzy Logic Controller – An Application of the Theory of Fuzzy Sets and Fuzzy Logic

6.1 Introduction

When the fuzzy logic systems described in Chapter 4 are used as controllers, they are called fuzzy logic controllers, abbreviated as FLC. Fuzzy logic control is by far the most successful application of the theory of fuzzy sets and fuzzy logic. Fuzzy logic control approach provides a systematic and efficient theoretical framework for incorporating linguistic descriptions of human expertise into design of automatic controllers. It has been shown [80, 177] that, under certain conditions, fuzzy systems are universal approximators, i.e., they are capable of approximating real continuous functions on a compact set, (i.e., on a closed and bounded set), to arbitrary accuracy. This provides an explanation for the practical success of fuzzy logic systems in engineering applications. It also indicates that, theoretically, it is possible to find an appropriate fuzzy logic controller for a large class of systems concerned.

In this chapter, a complete practical procedure for formulating expertise based

fuzzy logic controllers is presented via experiments and simulation examples. In the experiments, fuzzy logic controllers are designed to control a real flexible single link manipulator, described in Appendix B, whose dynamic model is unknown (or assumed so for purposes of controller design), where, when disturbed by external disturbances, (1) active damping of beam vibration and (2) position regulation of the tip of the beam are demonstrated. Their performances are also compared with that of conventional PD controllers. In the simulation example, a fuzzy logic controller is designed to command the tip of the flexible link of a manipulator to follow a predefined trajectory, where the flexible link manipulator is represented by a dynamic model. This model, however, is treated as a black box, and is unknown to the controller.

It should be emphasized that the resulting fuzzy logic controllers here are based solely on human expertise and trial and error tuning, without explicit theoretical analysis. This is to form a basis for further development in following chapters, where analytical techniques are introduced to adaptively tune FLS parameters to improve system performance and assure system stability.

6.2 Basics of Fuzzy Logic Controller Design

The design of a FLC is an *ad hoc* procedure based on its four components: fuzzification interface, fuzzy rule base, fuzzy inference engine and defuzzification interface [88, 177]. The design process may be divided into the following steps.

- determine input and output variables,
- design fuzzification interface,
- design fuzzy rule base,
- design fuzzy inference engine, i.e., choose decision making logic,

- design defuzzification interface.

The selection of input and output variables of a FLC depends on the particular problem and designer's experience.

For the fuzzification interface, as pointed out previously, we use the singleton fuzzifier throughout this work. For the convenience of computer implementation of control algorithms, we decompose each fuzzifier into two parts, a prefilter and a fuzzifier, as shown in fig.6.1. The prefilters, f_p, $p = 1, \ldots, P$, transform the input

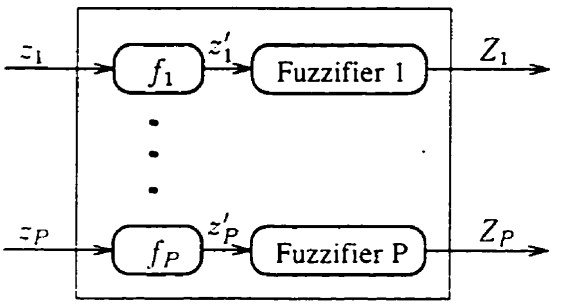


Figure 6.1: Reconfigured Fuzzification Interface

values of system states into desired ranges, i.e.,

$$z'_p = f_p(z_p) \triangleq (z_p - a_p) \cdot b_p \ . \tag{6.1}$$

where a_p and b_p are constants that translate and scale the values of input variables, respectively. We can by this means use a normalized fuzzy partition and adapt it to different input variables with different numerical regions by applying appropriate prefilters. For example, for an input variable, $z \in [5, 20]$, and a normalized fuzzy partition as shown in fig.6.2, where primary fuzzy sets are defined as {NB, NS, ZR, PS, PB}, we wish to transform the region of input values from [5, 20] to [-2.5, 2.5]. This can be achieved by applying a prefilter, f, defined as

$$z' \triangleq f(z) \triangleq (z - 12.5) \cdot \frac{2.5}{7.5} \ . \tag{6.2}$$

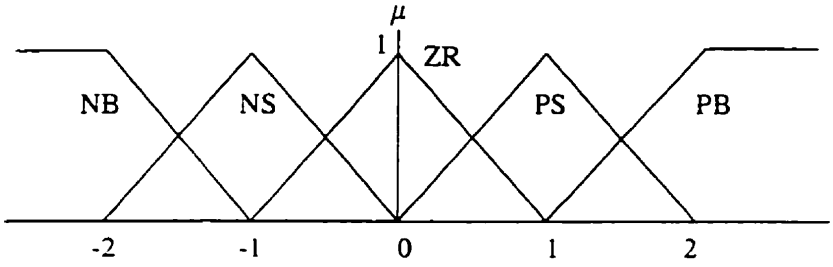


Figure 6.2: A Normalized Fuzzy Partition

The fuzzy rule base is the main part of an FLC. It is composed of linguistic control rules of the form (4.31), which map a set of fuzzy input sets to a fuzzy output set. The design of a fuzzy rule base may be decomposed into the following steps:

- fuzzy partition of linguistic variables, i.e., define primary fuzzy sets for every linguistic variable.

- define membership functions for every primary fuzzy set.

- define linguistic statements of fuzzy rules, i.e., define mappings from state space to output space.

The fuzzy inference engine is the decision making logic which, by using IF-THEN rules in the rule base, derives a fuzzy output set from fuzzy input sets. Its mathematical expression is given in eq.(4.12).

The defuzzification interface is a defuzzifier which transforms a fuzzy output set into a crisp control signal. For convenience in certain situations, a defuzzifier is sometimes decomposed into two components, a defuzzifier and a scale factor, as shown in fig.6.3.

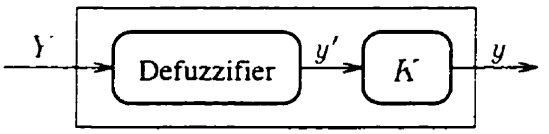


Figure 6.3: Reconfigured Defuzzifier

6.3 Fuzzy Logic Control of Flexible Link Manipulator: Design and Experimental Demonstrations

A flexible single link manipulator was built to serve as an experimental testbed, with details of this mechanical system presented in Appendix B. Although this is only a single degree of freedom system, the very flexible link, the presence of nonlinear joint friction, and existence of an unknown internal control loop make this manipulator a very challenging system for identification and control purposes.

6.3.1 Active Link Vibration Control with FLC

In this section, a fuzzy logic controller is developed to actively suppress the deflection at the tip of the manipulator link caused by external disturbances. The main objective here is to demonstrate the basic ideas and procedures for developing a fuzzy logic controller.

Stage 1 Accumulating Experience

Initially we have no knowledge of the dynamics of this manipulator. For FLC design, the first step is to acquire familiarity with the dynamic behavior of this mechanical system. To achieve this an operator manually controls the input voltage of the motor by operating a joystick to suppress the tip vibration of the link via eye-hand coordination, as illustrated in fig.6.4(a). No feedback sensors are used at this stage

of FLC development which is for a human operator to acquire intuition about system dynamic behavior and accumulate expertise in controlling the system.

Stage 2 Refining Expertise and Collecting Input-Output Data

At this stage, an ultrasonic position sensor is incorporated into the system, and tip deflection of the link is measured and displayed on a computer screen in real time. The rate of change of the tip deflection of the link is obtained through numerical differentiation of the tip deflection and is also displayed on the computer screen. Now, rather than looking at the real beam, the operator manipulates the joystick while watching the computer screen, where measured information is displayed, as illustrated in fig.6.4(b). This is nearer to the real automatic control situation where only sensed information is accessible by automatic controllers. In this stage, the expertise of the human operator is refined, and both input and output data are recorded, and can be processed and analysed off-line to help generate and tune IF-THEN rules.

Stage 3 Design of Automatic FLC

The next step is the design of the automatic fuzzy logic controller. We choose tip deflection of the link, $\epsilon \triangleq v(L,t) \in \mathcal{E}$, and its rate of change, $c \triangleq \dot{v}(L,t) \in \mathcal{C}$, as the inputs of the FLC, and the command voltage of the motor, $\delta \in \mathcal{D}$, as the output, where $\mathcal{E}, \mathcal{C}, \mathcal{D} \subset R$, are the universes of discourse for ϵ, c and δ, respectively. In addition,

$$c \triangleq \dot{v}(L,t) \triangleq \frac{v(L,t) - v(L,t-T)}{T} , \tag{6.3}$$

and T is the sampling period, which is 0.025 sec in this experiment. The control system is illustrated in fig.6.4(c), and the overall structure of the FLC is illustrated in fig.6.5.

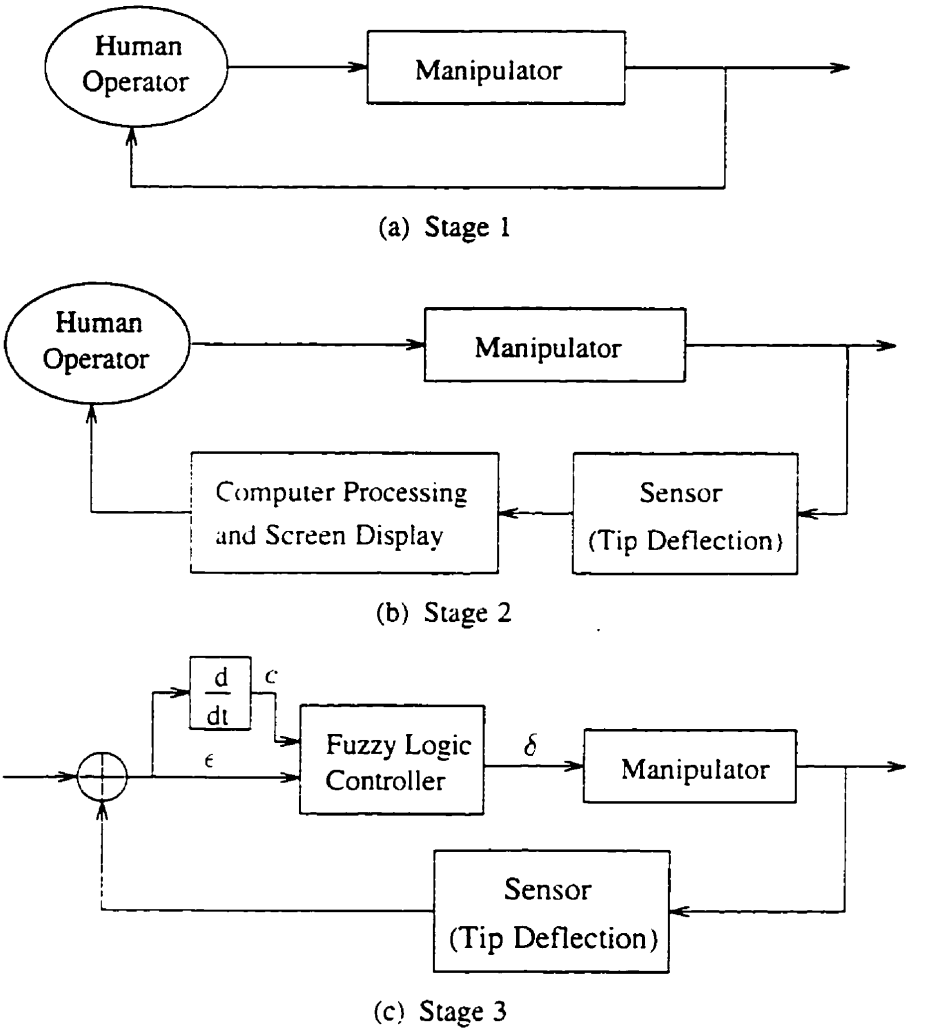


Figure 6.4: Fuzzy Logic Controller Development

Seven primary fuzzy sets are defined in $\mathcal{E}$, $\mathcal{C}$ and $\mathcal{D}$, respectively, i.e.,

$$\begin{cases} \mathcal{E} & : \quad \{E_1,\ E_2,\ E_3,\ E_4,\ E_5,\ E_6,\ E_7\}\ . \\ \mathcal{C} & : \quad \{C_1,\ C_2,\ C_3,\ C_4,\ C_5,\ C_6,\ C_7\}\ . \\ \mathcal{D} & : \quad \{\Delta_1,\ \Delta_2,\ \Delta_3,\ \Delta_4,\ \Delta_5,\ \Delta_6,\ \Delta_7\}\ . \end{cases} \tag{6.4}$$

For the notation to be linguistically meaningful, let

$$\begin{aligned} &\{E_1,\ E_2,\ E_3,\ E_4,\ E_5,\ E_6,\ E_7\}^T \\ &\qquad \triangleq \{E_{NB},\ E_{NM},\ E_{NS},\ E_{ZR},\ E_{PS},\ E_{PM},\ E_{PB}\}^T\ , \\ &\{C_1,\ C_2,\ C_3,\ C_4,\ C_5,\ C_6,\ C_7\}^T \end{aligned} \tag{6.5}$$

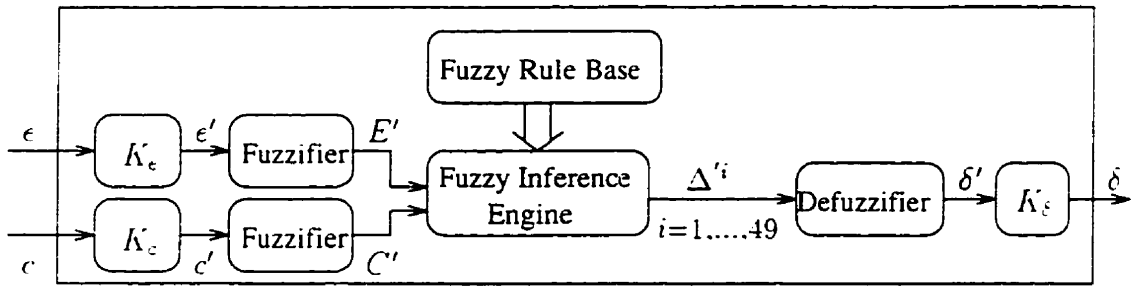


Figure 6.5: FLC for Vibration Control Experiment

$$\triangleq \{C_{NB},\ C_{NM},\ C_{NS},\ C_{ZR},\ C_{PS},\ C_{PM},\ C_{PB}\}^T\ . \tag{6.6}$$

$$\{\Delta_1,\ \Delta_2,\ \Delta_3,\ \Delta_4,\ \Delta_5,\ \Delta_6,\ \Delta_7\}^T$$

$$\triangleq \{\Delta_{NB},\ \Delta_{NM},\ \Delta_{NS},\ \Delta_{ZR},\ \Delta_{PS},\ \Delta_{PM},\ \Delta_{PB}\}^T\ . \tag{6.7}$$

where N - *Negative*, P - *Positive*, B - *Big*, M - *Medium*, S - *Small*. The numerical subscripts - 1,2,..... and the literal subscripts - NB, NM,..... will be used interchangeably - the former being more convenient in mathematical expressions while the latter is linguistically more appealing.

Gaussian type membership functions are used for each primary fuzzy set.

$$\begin{aligned}
\mu_{E_1}(\epsilon') &\triangleq \begin{cases} 1\ , & \text{if } \epsilon' < \bar{\epsilon}_1\ . \\ \exp[-\frac{1}{2}(\frac{\epsilon'-\bar{\epsilon}_1}{\sigma_{\epsilon_1}})^2]\ , & \text{if } \epsilon' \geq \bar{\epsilon}_1\ . \end{cases} \\
\mu_{E_j}(\epsilon') &\triangleq \exp[-\frac{1}{2}(\frac{\epsilon'-\bar{\epsilon}_j}{\sigma_{\epsilon_j}})^2]\ ,\ j = 2,3,....,6. \\
\mu_{E_7}(\epsilon') &\triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{\epsilon'-\bar{\epsilon}_7}{\sigma_{\epsilon_7}})^2]\ , & \text{if } \epsilon' < \bar{\epsilon}_7\ , \\ 1\ , & \text{if } \epsilon' \geq \bar{\epsilon}_7\ , \end{cases}
\end{aligned} \tag{6.8}$$

$$\begin{aligned}
\mu_{C_1}(c') &\triangleq \begin{cases} 1\ , & \text{if } c' < \bar{c}_1\ . \\ \exp[-\frac{1}{2}(\frac{c'-\bar{c}_1}{\sigma_{c_1}})^2]\ , & \text{if } c' \geq \bar{c}_1\ . \end{cases} \\
\mu_{C_j}(c') &\triangleq \exp[-\frac{1}{2}(\frac{c'-\bar{c}_j}{\sigma_{c_j}})^2]\ ,\ j = 2,3,....,6.
\end{aligned} \tag{6.9}$$

$$\mu_{C_7}(c') \triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{c'-\bar{c}_7}{\sigma_{c_7}})^2] . & \text{if } c' < \bar{c}_7 . \\ 1 . & \text{if } c' \geq \bar{c}_7 . \end{cases}$$

$$\mu_{\Delta_1}(\delta') \triangleq \begin{cases} 1 . & \text{if } \delta' < \bar{\delta}_1 . \\ \exp[-\frac{1}{2}(\frac{\delta'-\bar{\delta}_1}{\sigma_{\delta_1}})^2] . & \text{if } \delta' \geq \bar{\delta}_1 . \end{cases}$$

$$\mu_{\Delta_j}(\delta') \triangleq \exp[-\frac{1}{2}(\frac{\delta'-\bar{\delta}_j}{\sigma_{\delta_j}})^2] . \; j = 2, 3, \ldots, 6. \tag{6.10}$$

$$\mu_{\Delta_7}(\delta') \triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{\delta'-\bar{\delta}_7}{\sigma_{\delta_7}})^2] . & \text{if } \delta' < \bar{\delta}_7 . \\ 1 . & \text{if } \delta' \geq \bar{\delta}_7 . \end{cases}$$

where ϵ' and c' are transformed values of ϵ and c, respectively. For j=1,2,....7, σ_{ϵ_j}, σ_{c_j} and σ_{δ_j} are parameters characterizing the shape of the membership functions, and $\bar{\epsilon}_j$, $\bar{c}_j$ and $\bar{\delta}_j$ are parameters characterizing the locations of the corresponding primary fuzzy sets. The locations of the primary fuzzy sets are defined as

$$\begin{cases} \bar{\epsilon}_1 = \bar{c}_1 = \bar{\delta}_1 \hat{\equiv} -6 . \quad \bar{\epsilon}_2 = \bar{c}_2 = \bar{\delta}_2 \hat{\equiv} -4 . \quad \bar{\epsilon}_3 = \bar{c}_3 = \bar{\delta}_3 \hat{\equiv} -2 . \\ \bar{\epsilon}_4 = \bar{c}_4 = \bar{\delta}_4 \hat{\equiv} 0 . \\ \bar{\epsilon}_5 = \bar{c}_5 = \bar{\delta}_5 \hat{\equiv} 2 . \quad \bar{\epsilon}_6 = \bar{c}_6 = \bar{\delta}_6 \hat{\equiv} 4 . \quad \bar{\epsilon}_7 = \bar{c}_7 = \bar{\delta}_7 \hat{\equiv} 6 . \end{cases} \tag{6.11}$$

The shape of the primary fuzzy sets are defined as,

$$\sigma_{\epsilon_j} = \sigma_{c_j} = \sigma_{\delta_j} \triangleq \begin{cases} 0.3 . & for\ j = 4 . \\ 0.6 . & otherwise . \end{cases} \tag{6.12}$$

Figure 6.6 shows the primary fuzzy sets for variable ϵ.

The fuzzy rules are developed through operator's intuition and experience, and are further tuned through trial and error tests. This FLC has 49 rules, as shown in table 6.1. The table is read as

IF (ϵ is E_{NB} and c is C_{NB}) THEN (δ is Δ_{NB}) :

$$\vdots$$

We use the FLS-I mapping, as defined in section 4.6.1, for the FLC, which is characterized by the following properties, repeated here for convenience:

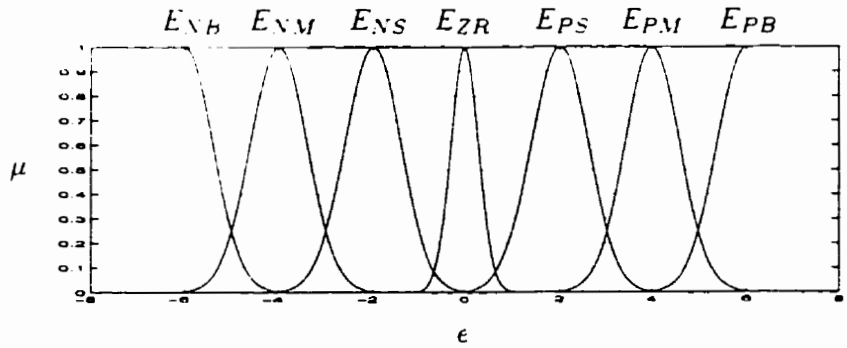


Figure 6.6: Primary Fuzzy Sets for ϵ in Vibration Control Experiment with FLC

Table 6.1: Fuzzy Rule Base of the FLC for Vibration Control Experiment

D \ C / $\mathcal{E}$	C_{NB}	C_{NM}	C_{NS}	C_{ZR}	C_{PS}	C_{PM}	C_{PB}
E_{NB}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{ZR}	Δ_{ZR}
E_{NM}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{ZR}	Δ_{ZR}
E_{NS}	Δ_{NM}	Δ_{NM}	Δ_{NM}	Δ_{NM}	Δ_{ZR}	Δ_{PS}	Δ_{PS}
E_{ZR}	Δ_{NM}	Δ_{NM}	Δ_{NS}	Δ_{ZR}	Δ_{PS}	Δ_{PM}	Δ_{PM}
E_{PS}	Δ_{NS}	Δ_{NS}	Δ_{ZR}	Δ_{PM}	Δ_{PM}	Δ_{PM}	Δ_{PM}
E_{PM}	Δ_{ZR}	Δ_{ZR}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{PB}
E_{PB}	Δ_{ZR}	Δ_{ZR}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{PB}

- singleton fuzzifier, eq.(4.19);
- "min" operation for fuzzy relations, eq.(4.20);
- "min" operation for fuzzy implication, eq.(4.21);
- "sup-min" compositional rule of fuzzy inference, eq.(4.22);
- CM defuzzifier, eqs.(4.27–4.28).

The crisp output, δ', of this FLC is given by eq.(4.27) as

$$\delta' \triangleq \frac{\sum_{i=1}^{I} \mu_{\Delta'^i}(\overline{\delta}^i) \cdot \overline{\delta}^i}{\sum_{i=1}^{I} \mu_{\Delta'^i}(\overline{y}^i)} \ . \tag{6.13}$$

where Δ'^i is the output fuzzy set induced by the ith rule, and $\mu_{\Delta'^i}(\overline{\delta}^i)$ is defined in eq.(4.26) as

$$\mu_{\Delta'^i}(\overline{\delta}^i) = \min\{\mu_{E'^i}(\epsilon'),\ \mu_{C'^i}(c'),\ \mu_{\Delta^i}(\overline{\delta}^i)\} \ . \tag{6.14}$$

where, for $i = 1, \ldots, 49$,

$$\begin{cases} E'^i \in \{E_j,\ j = 1, \ldots, 7\} \ . \\ C'^i \in \{C_j,\ j = 1, \ldots, 7\} \ . \\ \Delta^i \in \{\Delta_j,\ j = 1, \ldots, 7\} \ . \end{cases} \tag{6.15}$$

Equation (6.14) may be graphically illustrated as in fig.6.7, where the shaded area represents the output fuzzy set, Δ'^i induced by the ith rule, and $\overline{\delta}^i$ is defined in eq.(4.28) as

$$\overline{\delta}^i \triangleq \min_{\delta'^i \in \Delta'^i} \{\ |\delta'^i| :\ \delta'^i = arg \sup_{\delta'^i \in D} \{\mu_{\Delta'^i}(\delta')\}\ \} \ . \tag{6.16}$$

which is also illustrated in fig.(6.7). The $\overline{\delta}^i$ is selected according to the following rationale: among the infinite number of support points in Δ'^i, we take only that which has associated with it the maximum grade of membership, and in the case there are several such support points, we take that with the least absolute value.

For ϵ in cm and c in cm/sec, let

$$\begin{cases} K_\epsilon \triangleq 1.6 \ . \\ K_c \triangleq 0.04 \ . \\ K_\delta \triangleq 6.7 \ . \end{cases} \tag{6.17}$$

Figure 6.8 shows the free vibration at the tip of the link, $v(L,t)$, when disturbed by an external disturbance (an impact). Figure 6.9 illustrates a vibration control experiment when the initially still link is subjected to external disturbances. Figure

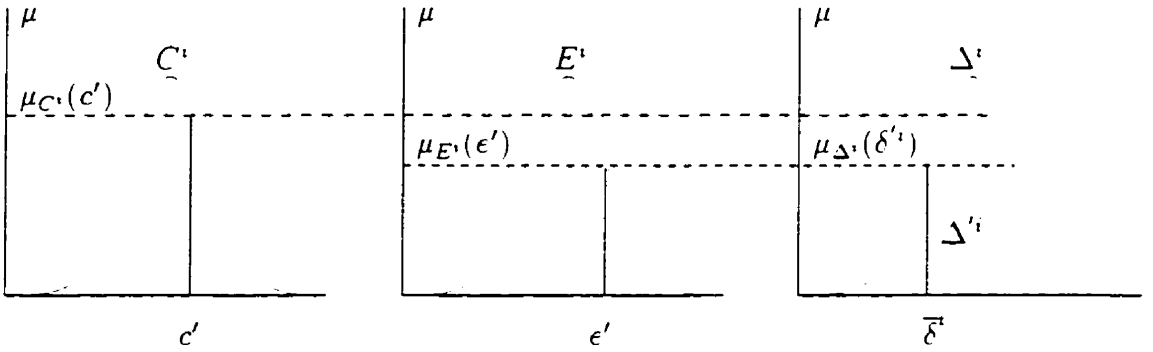


Figure 6.7: Illustration of Δ'' and $\overline{\delta''}$

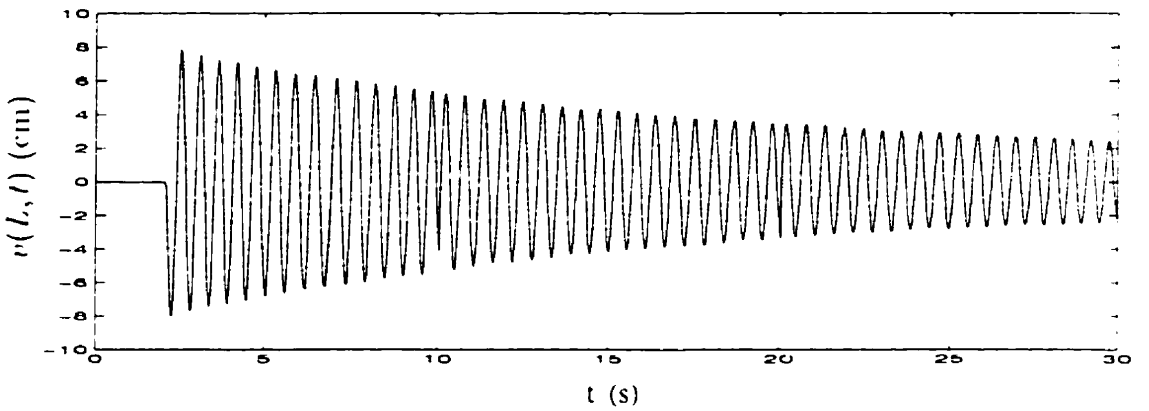
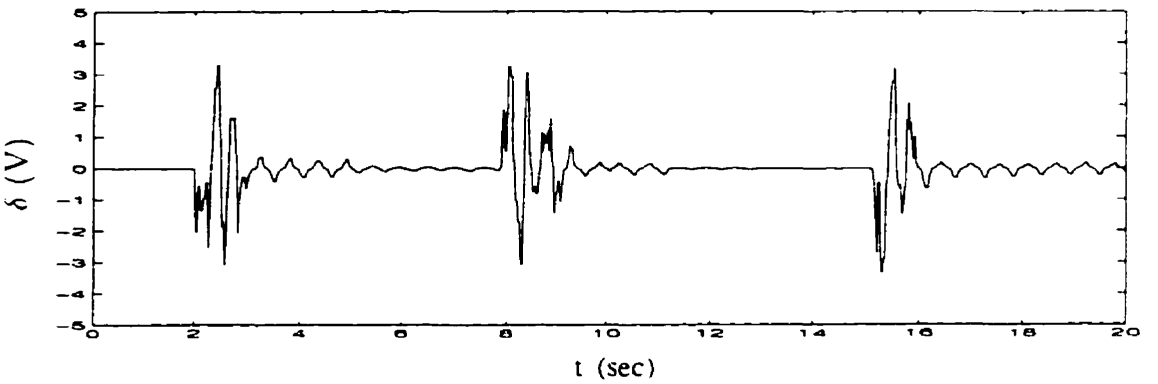


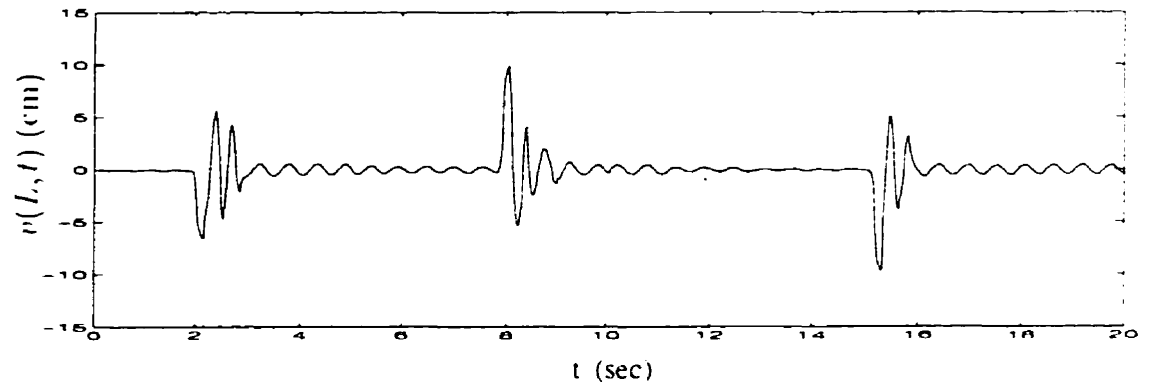
Figure 6.8: Free Vibration at the Tip of the Link When Disturbed

6.9(a) shows the command voltage of the motor, δ, produced by the FLC, and figure 6.9(b) shows the vibrational displacement, $v(L,t)$, at the tip of the link. The disturbances are imposed at $t \approx 2$ sec, 8 sec, and 15 sec.

It is seen that the tip vibration of the link can be quickly damped out. The usefulness and effectiveness of a fuzzy logic controller is clearly demonstrated. There does remain a small residual vibration with magnitude about 1 cm. This small residual vibration is very hard to be suppressed actively mainly due to the nonlinear friction of the motor.



(a) Command Voltage of the Motor with FLC



(b) Vibrational Displacement at the Tip of the Link
(Disturbances Presented at $t \approx 2$ sec, 8 sec, and 15 sec)

Figure 6.9: Active Vibration Control Experiment with FLC

6.3.2 Vibration and Position Control with FLC

Above, we have used a FLC to actively control only the tip deflection of the beam, with the angular position of the link of no concern, so that the final position of the link can be different from its original position. We next consider a more realistic control task, which is still to actively suppress the beam vibration due to external disturbances, but at the same time, the beam is not allowed to deviate from its original angular position.

Define state variables,

$$\begin{cases} e(t) \triangleq v(L,t) - K_u^e \cdot o(0,t) , \\ c(t) \triangleq \dot{v}(L,t) - K_u^c \cdot \dot{o}(0,t) , \end{cases} \tag{6.18}$$

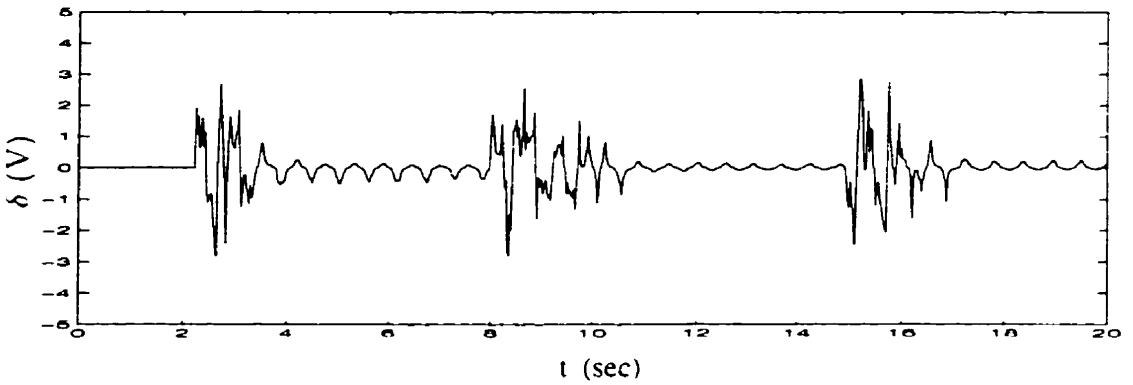
where K_u^e and K_u^c are the weighting coefficients for $o(0,t)$ and $\dot{o}(0,t)$, respectively. $v(L,t)$ and $\dot{v}(L,t)$ are the vibrational displacement and velocity, respectively, at the tip of the beam. $o(0,t)$ and $\dot{o}(0,t)$ are the angular displacement and velocity of the motor hub. $v(L,t)$ is measured with the ultrasonic position sensor, and $\dot{v}(L,t)$ is obtained through numerical differentiation of $v(L,t)$, as given in eq.(6.3), where the sampling period, T, is 0.025 sec. $o(0,t)$ and $\dot{o}(0,t)$ are measured with built in encoder and velometer, respectively.

The same FLC developed in the vibration control experiment, section 6.3.1, is used here for this vibration and position control experiment. Here, ϵ and c defined in eq.(6.18) are the inputs of the FLC, and the output of the FLC, δ, is the command voltage of the motor. For $v(L,t)$ in cm, $\dot{v}(L,t)$ in cm/sec, $o(0,t)$ in degrees, and $\dot{o}(0,t)$ in rad/sec, the weighting parameters and the gains of the FLC are defined as

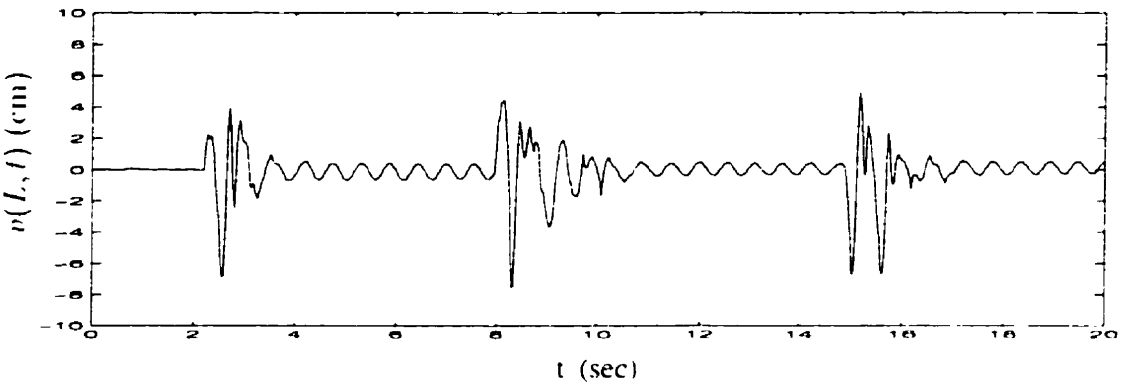
$$\begin{cases} K_u^e \triangleq 2.5 , \\ K_u^c \triangleq 30 , \\ K_e \triangleq 1.6 , \\ K_c \triangleq 0.04 , \\ K_\delta \triangleq 6 . \end{cases} \tag{6.19}$$

A vibration and position control experiment with the FLC is illustrated in fig.6.10(a-c), where external disturbances presented at $t \approx 2$ sec, 8 sec, and 15 sec, which are imposed by manually hitting the beam. Figure 6.10(a) shows the control signal, 6.10(b) shows the vibrational displacement at the tip of the link, and 6.10(c) shows the angular displacement of the motor hub.

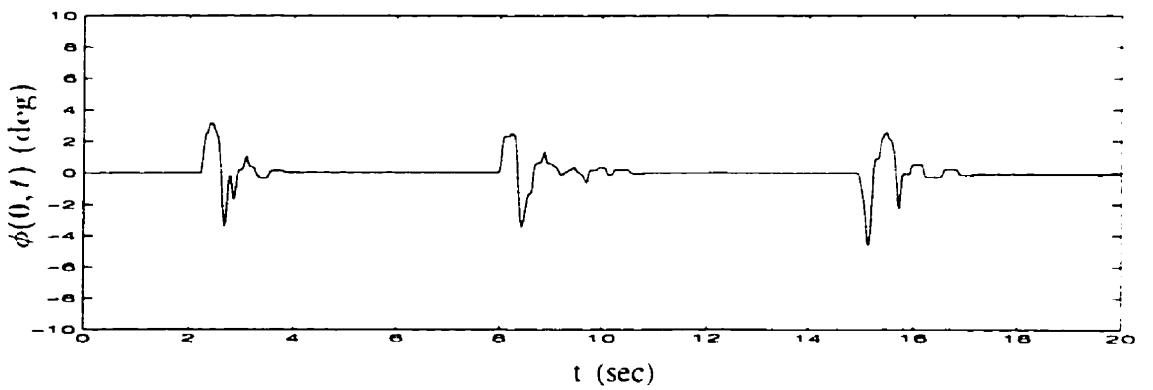
It is seen that the tip vibration of the beam is effectively suppressed, and at the same time, the angular position of the beam is also maintained at its original setting, which, again, demonstrates the capability of the fuzzy logic control approach.



(a) Command Voltage of the Motor with FLC



(b) Vibrational Displacement at the Tip of the Link



(c) Angular Displacement at the Hub of the Motor

(Disturbances Presented at $t \approx 2$ sec, 8 sec, and 15 sec)

Figure 6.10: Experiment on Vibration and Position Control with FLC

6.4 Experiments with PD Control

6.4.1 Introduction

The proportional-integral-derivative (PID) control approach is widely used in industrial applications. Process engineers and operation personnel are familiar with it, and most industrial single-loop processes are robustly and stably controlled with it [58, Chap.4]. Although this controller is not designed as optimal, it is estimated that more sophisticated controllers would not confer an economic advantage over PID in 80% of control loops [24]. The PID control approach can also be model free, as is the fuzzy logic control approach. A natural question raised here, therefore, is how the performance of a fuzzy logic controller compares with that of a PID controller? In the following, we use PD controllers to repeat the experiments presented in the last section with FLC - that is, to first actively suppress beam vibration, and second, to control the beam vibration and maintain the beam angular position at the same time.

6.4.2 Active Link Vibration Control with PD Controller

The system with PD controller is shown in fig.6.11, where the output of the PD controller, δ, is the command voltage of the motor and is obtained as

$$\delta \triangleq K_p\epsilon + K_d c \ , \tag{6.20}$$

where as previously, $\epsilon \triangleq v(L,t)$ and $c \triangleq \dot{\epsilon}$ are the vibrational deflection at the tip of the beam and its time rate of change, respectively. Again, c is obtained with numerical differentiation of ϵ, as given in eq.(6.3), where the sampling period, T, is 0.025 sec. The objective is to actively suppress the beam vibration due to external disturbances.

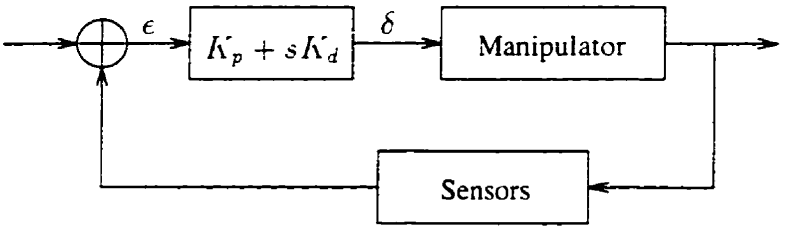


Figure 6.11: PD Control System For Vibration Control Experiment

For ϵ in cm and c in cm/sec. gains are selected. by trial and error. to be:

$$\begin{cases} K_p \triangleq 0.27 \ . \\ K_d \triangleq 0.015 \ . \end{cases} \tag{6.21}$$

A vibration control experiment is illustrated in fig.6.12(a-b). where external disturbances are imposed at $t \approx 2$ sec. 8 sec. and 15 sec. Figure 6.12(a) shows the control signal. δ. and 6.12(b) shows the vibrational displacement. ϵ. at the tip of the link. Comparing fig.6.12 to the results of the similar experiment with FLC. fig.6.9. it is found that both types of controllers perform well. but the FLC appears to have an edge in the residual vibration.

6.4.3 Vibration and Position Control with PD Controller

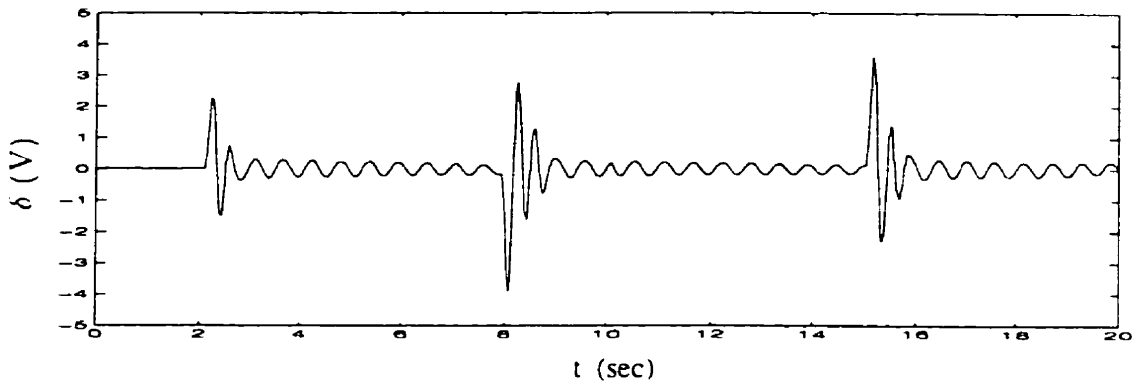
Now. we consider an experiment similar to that presented in section 6.3.2 with the FLC. which is to actively suppress the beam vibration due to external disturbances. but at the same time. the beam is not to deviate from its original angular position.

Define state variables.

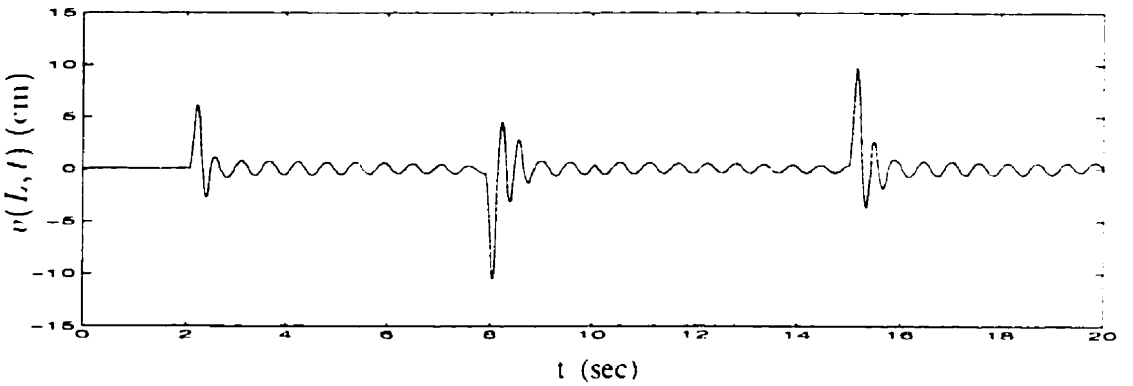
$$\begin{cases} \epsilon(t) \triangleq v(L,t) - K^{\epsilon} \cdot o(0,t) \ . \\ c(t) \triangleq \dot{v}(L,t) - K^{c} \cdot \dot{o}(0,t) \ . \end{cases} \tag{6.22}$$

where K^{ϵ} and K^{c} are the weighting coefficients for $o(0,t)$ and $\dot{o}(0,t)$. respectively.

The PD controller illustrated in fig.6.11 is used for this vibration and position



(a) Command Voltage of the Motor with PD Controller



(b) Vibrational Displacement at the Tip of the Link
(Disturbances Presented at $t \approx 2$ sec, 8 sec, and 15 sec)

Figure 6.12: Active Vibration Control Experiment with PD Controller

control experiment, and the output of the PD controller, δ, is obtained from

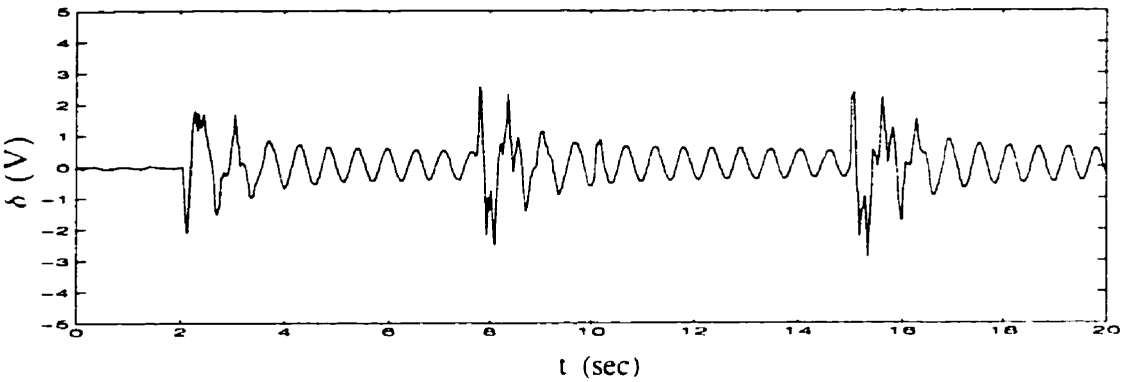
$$\delta \triangleq K_p \epsilon + K_d c \ . \tag{6.23}$$

with ϵ and c here being defined in eq.(6.22). For $v(L,t)$ in cm, $\dot{v}(L,t)$ in cm/sec, $o(0,t)$ in degree, $\dot{o}(0,t)$ in rad/sec, the weighting parameters and the gains of the PD controller are:

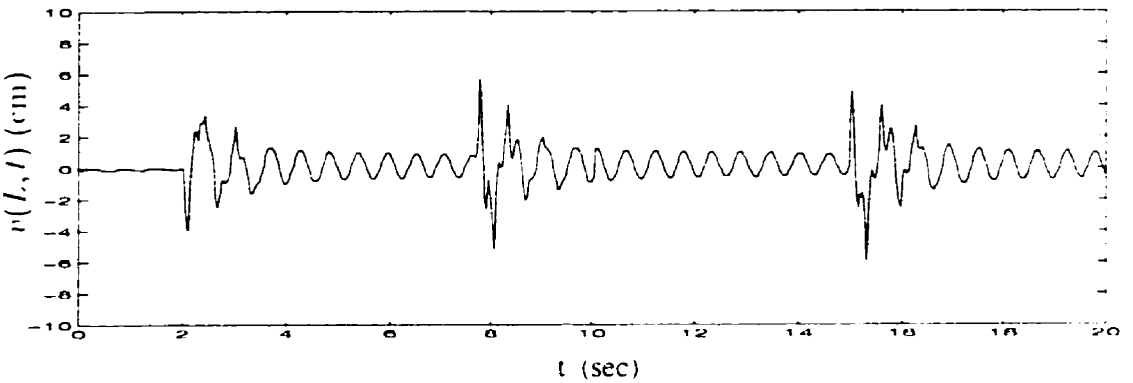
$$\begin{cases} K^{\epsilon} \triangleq 2.5 \ . \\ K^{c} \triangleq 15 \ . \\ K_p \triangleq 0.45 \ . \\ K_d \triangleq 0.02 \ . \end{cases} \tag{6.24}$$

A vibration and position control experiment with the PD controller is illustrated in fig.6.13(a–c), where external disturbances are imposed at $t \approx 2$ sec, 8 sec, and 15 sec. Figure 6.13(a) shows the control signal, 6.13(b) shows the vibrational displacement at the tip of the link, and 6.13(c) shows the angular displacement of the motor hub.

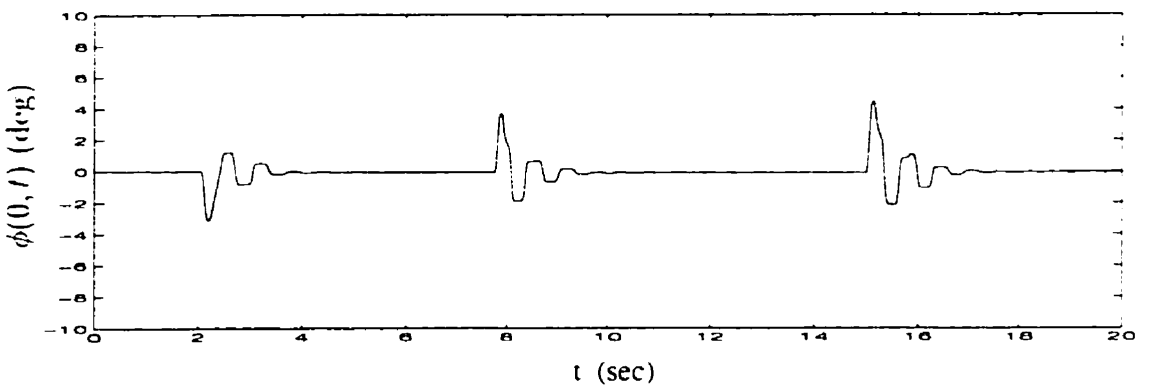
Comparing fig.6.13 to the results of the similar experiment with FLC, fig.6.10, it is again observed that both the FLC and PD perform well, but the residual vibration of the beam with FLC is smaller than that with PD controller.



(a) Command Voltage of the Motor with PD Controller



(b) Vibrational Displacement at the Tip of the Link



(c) Angular Displacement at the Hub of the Motor

(Disturbances Presented at $t \approx 2$ sec, 8 sec, and 15 sec)

Figure 6.13: Experiment on Vibration and Position Control with PD Controller

6.4.4 Remarks

- Although in the simple control tasks presented above, the FLC approach is only slightly better than that of PD control approach, for more complicated situations such as industrial process controls [10, 169, 210], human expertise can be of much more significance for successful control, when the simple PID control approach may not work at all.

- The design procedures of FLC and PID control are fundamentally different. The PID control approach is basically a trial and error approach, while the FLC approach provides a systematic framework to explicitly incorporate human expertise in the form of IF-THEN rules. The fuzzy IF-THEN rules can be rationally set up and modified, as opposed to trial and error for obtaining the gains in the case of PID control.

- The fuzzy logic control approach has the potential to be further developed into an effective approach in dealing with highly nonlinear and ill-defined systems, where the overall system stability properties and performance criteria can be theoretically analysed and adaptively adjusted, as will be seen later in this work.

- From our experience, the FLC appears more stable and robust than PD controller. The former is less sensitive to the values of gains, change of the tip mass, and the strength of external disturbances.

- In the literature, it has been reported [33, 102] that fuzzy logic controllers are more robust than PID controllers, since they can cover much wider ranges of operating conditions than can PID controllers; developing a FLC is frequently cheaper than developing a model based or other controller with equivalent performance; fuzzy logic controllers are customizable, since it is easy to un-

derstand and modify their rules, which not only mimic a human operator's strategies but are also expressed in linguistic terms in natural language. Our observations are generally in accord with these views.

6.5 Trajectory Control of a Flexible Link Manipulator – A Simulation Example with FLC

Consider the dynamic model of a flexible link manipulator given in eq.(B.4) with motor friction neglected,

$$\begin{cases} \dot{X} &= AX + BK_T V_m \ , \\ Y &= CX \ , \end{cases} \tag{6.25}$$

where K_T is the motor torque constant, V_m is the command voltage of the motor, X and Y are given in eq.(A.137), A, B and C are given in eqs.(A.141-A.143). This models the torque control mode of the direct drive motor described in Appendix B.

The objective here is to develop a FLC to command the tip of the link of this manipulator to follow a desired trajectory. Simulation results will be presented, but the dynamic model of the manipulator is treated strictly as a black box for the FLC, i.e., it is unknown to the FLC.

Let the desired tip trajectory of the link be $w_d(L,t)$, and the actual trajectory of the tip of the link be $w(L,t)$. The difference between these is

$$\epsilon(t) \triangleq w(L,t) - w_d(L,t) \ . \tag{6.26}$$

Let the rate of change of $\epsilon(t)$ be $c(t)$, i.e.

$$c(t) \triangleq \dot{\epsilon}(t) = \dot{w}(L,t) - \dot{w}_d(t) \ . \tag{6.27}$$

The overall system is illustrated by a block diagram in fig.6.14, where bold lines represent vector signal flows.

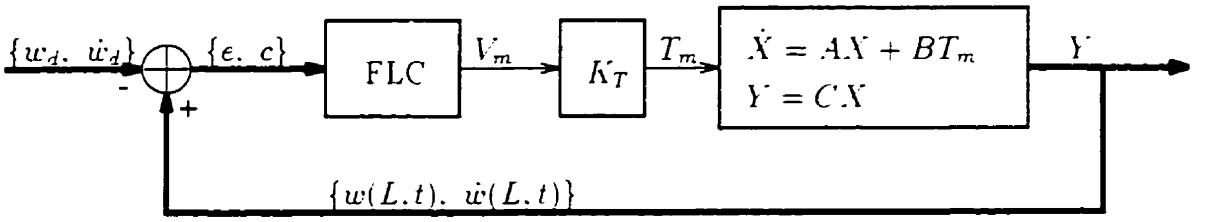


Figure 6.14: Control System Diagram for Trajectory Control with FLC

We choose $\epsilon \in \mathcal{E} \subset R$ and $c \in \mathcal{C} \subset R$ as the inputs of the FLC. The output of the FLC is the command voltage, $\delta \triangleq V_m \in \mathcal{D} \subset R$, where $\mathcal{E}$, $\mathcal{C}$ and $\mathcal{D}$ are the respective universes of discourse. We use the same FLC as developed in the active vibration control experiment of section 6.3.1, illustrated in fig.6.5, except with different parameter values here.

The locations of the primary fuzzy sets are defined as

$$\begin{cases} \bar{\epsilon}_1 = \bar{c}_1 = \bar{\delta}_1 \triangleq -3\,, \quad \bar{\epsilon}_2 = \bar{c}_2 = \bar{\delta}_2 \triangleq -2\,, \quad \bar{\epsilon}_3 = \bar{c}_3 = \bar{\delta}_3 \triangleq -1\,, \\ \bar{\epsilon}_4 = \bar{c}_4 = \bar{\delta}_4 \triangleq 0\,, \\ \bar{\epsilon}_5 = \bar{c}_5 = \bar{\delta}_5 \triangleq 1\,, \quad \bar{\epsilon}_6 = \bar{c}_6 = \bar{\delta}_6 \triangleq 2\,, \quad \bar{\epsilon}_7 = \bar{c}_7 = \bar{\delta}_7 \triangleq 3\,. \end{cases} \tag{6.28}$$

For want of a compelling reason to do otherwise, and for simplicity, the same shape of membership function was used for all of the primary fuzzy sets, by assigning

$$\sigma_{\epsilon_j} = \sigma_{c_j} = \sigma_{\delta_j} \triangleq 0.45\,, \quad j = 1, 2, \ldots, 7\,. \tag{6.29}$$

Figure 6.15 shows the primary fuzzy sets for variable ϵ defined above. As before, the fuzzy rules were developed through an operator's intuition and experience. The 49 rules of this FLC are summarized in table 6.2.

The flexible link is of stainless steel plate of 700 mm long, 1.27 mm thick and 50 mm wide, with Young's modulus $E = 206.7 \times 10^9$ N/m^2 and mass density $\rho = 7.89 \times 10^3$ Kg/m^3. The tip of the link is subjected to a payload of the same mass as the link. The structural damping coefficients of the flexural modes are assumed to be 0.002, i.e., $\xi_i \triangleq 0.002$, and the motor torque constant is taken to be $K_T \triangleq 18.75$

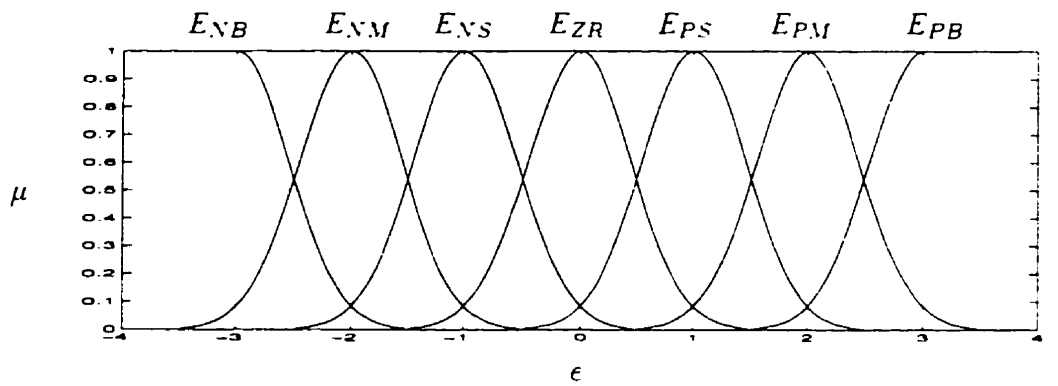


Figure 6.15: Primary Fuzzy Sets for ϵ in Trajectory Control Example with FLC

Table 6.2: Fuzzy Rule Base of the FLC for Trajectory Control

$\mathcal{D}$ \ $\mathcal{C}$ / $\mathcal{E}$	C_{NB}	C_{NM}	C_{NS}	C_{ZR}	C_{PS}	C_{PM}	C_{PB}
E_{NB}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{ZR}	Δ_{ZR}
E_{NM}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{PB}	Δ_{ZR}	Δ_{ZR}
E_{NS}	Δ_{PM}	Δ_{PM}	Δ_{PM}	Δ_{PM}	Δ_{ZR}	Δ_{NS}	Δ_{NS}
E_{ZR}	Δ_{PM}	Δ_{PM}	Δ_{PS}	Δ_{ZR}	Δ_{NS}	Δ_{NM}	Δ_{NM}
E_{PS}	Δ_{PS}	Δ_{PS}	Δ_{ZR}	Δ_{NM}	Δ_{NM}	Δ_{NM}	Δ_{NM}
E_{PM}	Δ_{ZR}	Δ_{ZR}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{NB}
E_{PB}	Δ_{ZR}	Δ_{ZR}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{NB}	Δ_{NB}

N·m/V. The dynamic model of the manipulator was implemented in SIMULINK, and the simulation programmed in MATLAB.

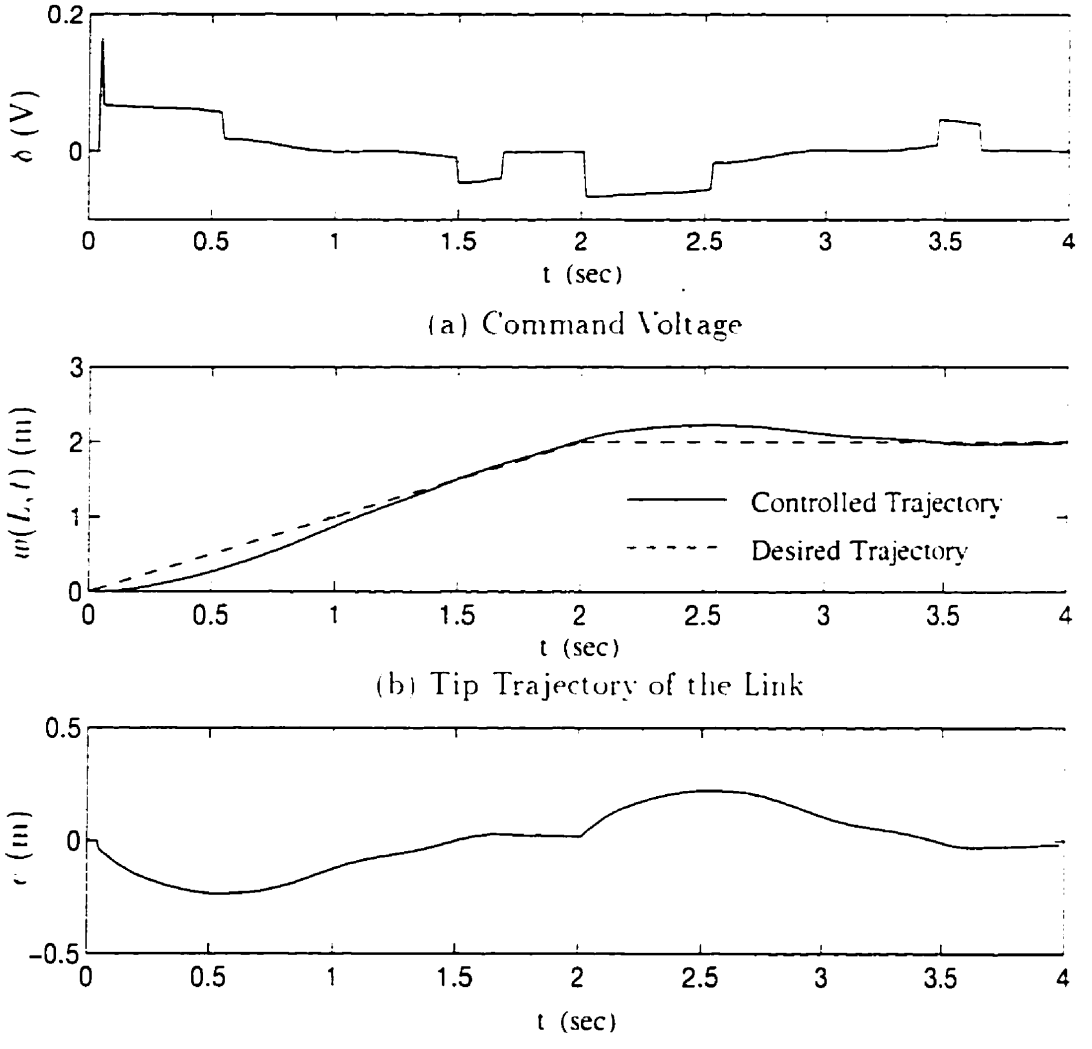
The sampling rate was set at 100 Hz, and,

$$\begin{cases} K_e \triangleq 0.07\,, \\ K_c \triangleq 0.07\,, \\ K_s \triangleq 10\,. \end{cases} \tag{6.30}$$

A simulation experiment for trajectory control of the manipulator is shown in fig.6.16, where 6.16(a) shows the control signal, 6.16(b) shows both the desired

trajectory and the controlled trajectory, and 6.16(c) shows the difference between the desired trajectory and the controlled trajectory.

The results are quite satisfactory, which demonstrates once again the effectiveness of the human knowledge based FLC, which does not utilize the system mathematical model.



(a) Command Voltage

(b) Tip Trajectory of the Link

(c) Tracking Error

Figure 6.16: Trajectory Control of a Flexible Link Manipulator

6.6 Concluding Remarks

In addition to the remarks made in section 6.4.4, we have following remarks and observations.

- In this chapter, we have applied the theory of fuzzy sets and fuzzy logic to control applications, and presented a process for formulating expert knowledge based fuzzy logic controllers. We have demonstrated applications of those fuzzy logic controllers for various control tasks, both experimentally and numerically. We also compared the performance of FLC with that of PD controllers.

- In FLC design, human expertise can be systematically incorporated in the form of IF-THEN rules.

- In both experimental demonstrations and simulation examples, the fuzzy logic controllers designed solely on the basis of expert knowledge demonstrate the capability of performing various control tasks satisfactorily, without knowledge of system mathematical models.

- In this chapter, FLC parameters must be designed *ad hoc* depending on individual systems and control tasks, and must sometimes be adjusted manually by trial and error. There is no formal synthesis which can theoretically guarantee system stability and acceptable performance. This is a major weakness of this FLC design scheme.

Later in this work, we present a new fuzzy logic system structure which can be more effective in dealing with dynamic systems. New identification and control algorithms are also developed, whose stability properties and performance criteria can be theoretically justified and parameters adaptively adjusted.

Chapter 7

Optimal Training For Fuzzy Logic Systems

7.1 Preliminary

In the previous chapter we presented an empirical fuzzy logic system (FLS) design process and its application in automatic control, in which all FLS parameters are obtained either by expert knowledge or by trial and error. In this chapter, we present an optimal training approach for FLS parameters by using a least-squares estimation technique and backpropagation algorithm.

As pointed out in chapter 2, least-square estimation and backpropagation techniques have been used in tuning FLS parameters. But it is also known that backpropagation approach has such weaknesses as generally being slow in convergence rate and the possibility of being trapped in local minima, while the least-square estimation technique suffers from the restriction of a rank condition over a parameter matrix. These motivated us to propose a training scheme that combines the both mentioned techniques in such a way that the new scheme would include the advantages but avoid the weaknesses of the both ingredient techniques.

Consider a FLS with input variables $z_p \in \mathcal{Z}_p \subset R$, $p = 1, \ldots, P$, where this

FLS is characterized by singleton fuzzifiers, CM defuzzifier, the product implication rule, sup-product compositional operator, Gaussian type membership functions for primary fuzzy sets, and algebraic product operation for all the "$\star$" operators. Its output, y, is given in eq.(4.45) as

$$y = \mathbf{\Theta}^T(\mathbf{z})\overline{\mathbf{Y}} \ . \tag{7.1}$$

where $\mathbf{z} \triangleq \{z_1, \ldots, z_P\}^T$, $\mathbf{\Theta}(\mathbf{z})$ and $\overline{\mathbf{Y}}$ are defined in eq.(4.44) as

$$\begin{cases} \mathbf{\Theta}(\mathbf{z}) & \triangleq & \{\theta_1(\mathbf{z}), \ldots, \theta_I(\mathbf{z})\}^T \ . \\ \overline{\mathbf{Y}} & \triangleq & \{\overline{y}_1, \ldots, \overline{y}_I\}^T \ . \end{cases} \tag{7.2}$$

$\theta_i(\mathbf{z})$, $i = 1, \ldots, I$, are given in eq.(4.43) as,

$$\theta_i(\mathbf{z}) \triangleq \frac{\prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{z_p - \Xi_p^i}{\sigma_p^i})^2]}{\sum_{i=1}^{I} \prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{z_p - \Xi_p^i}{\sigma_p^i})^2]} \ . \tag{7.3}$$

where Ξ_p^i and σ_p^i are design parameters that characterize the centers and shapes of membership functions of primary fuzzy sets. In the following discussions, we assume that the number of fuzzy rules, I, has been defined, leaving Ξ_p^i, σ_p^i and $\overline{y}_i$, $i = 1, \ldots, I$, as free parameters.

The problem is to determine parameters Ξ_p^i, σ_p^i and $\overline{y}_i$ for given input-output data pairs, $(\mathbf{z}(k), q(k))$, $k = 1, \ldots, N$, $\mathbf{z}(k) \triangleq \{z_1(k), \ldots, z_P(k)\}^T \in \mathcal{Z} \subset R^P$, $\mathcal{Z} \triangleq \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_P$, $q(k) \in R$, such that the loss functional - sum of the squared errors between the observations and the model outputs:

$$J_N \triangleq \sum_{k=1}^{N} E_k \triangleq \frac{1}{2} \sum_{k=1}^{N} [\ y(k) - q(k)\ ]^2 = \frac{1}{2} \| \ Y_N - Q_N \ \|^2 \ . \tag{7.4}$$

is minimized, where $q(k)$ is the model output, and $y(k)$ is the FLS output, both at time step k, and

$$\begin{cases} y(k) & \triangleq & \mathbf{\Theta}^T(k)\overline{\mathbf{Y}}(k) \ . \\ E_k & \triangleq & \frac{1}{2}[\ y(k) - q(k)\ ]^2 \ . \\ Y_N & \triangleq & \{y(1), \ldots, y(N)\}^T \ . \\ Q_N & \triangleq & \{q(1), \ldots, q(N)\}^T \ . \end{cases} \tag{7.5}$$

7.2 Optimal Training for FLS

7.2.1 Backpropagation Training

In the literature, the backpropagation algorithm (BP) has been used to train the parameters $\overline{z}_p^i$, σ_p^i, and $\overline{y}_i$ of the FLS in eq.(7.1) [177]. This approach causes the loss functional to decrease in the direction of steepest descent. The training rules are as follows, for $i = 1, \ldots, I$, $p = 1, \ldots, P$.

$$\overline{y}_i(k+1) = \overline{y}_i(k) - \alpha_{\overline{y}_i} \frac{\partial J_k}{\partial \overline{y}_i} |_{\overline{y}_i = \overline{y}_i(k)} , \tag{7.6}$$

$$\overline{z}_p^i(k+1) = \overline{z}_p^i(k) - \alpha_{\overline{z}_p^i} \frac{\partial J_k}{\partial \overline{z}_p^i} |_{\overline{z}_p^i = \overline{z}_p^i(k)} , \tag{7.7}$$

$$\sigma_p^i(k+1) = \sigma_p^i(k) - \alpha_{\sigma_p^i} \frac{\partial J_k}{\partial \sigma_p^i} |_{\sigma_p^i = \sigma_p^i(k)} , \tag{7.8}$$

where $\alpha_{\overline{y}_i}$, $\alpha_{\overline{z}_p^i}$ and $\alpha_{\sigma_p^i}$ are the learning rates of the respective variables. For convenience in on-line applications, the gradient descent is usually based on E_k of (7.5) rather than an J_k. Accordingly, the training rules are modified as [177].

$$\overline{y}_i(k+1) = \overline{y}_i(k) - \alpha_{\overline{y}_i} \frac{\partial E_k}{\partial \overline{y}_i} |_{\overline{y}_i = \overline{y}_i(k)} , \tag{7.9}$$

$$\overline{z}_p^i(k+1) = \overline{z}_p^i(k) - \alpha_{\overline{z}_p^i} \frac{\partial E_k}{\partial \overline{z}_p^i} |_{\overline{z}_p^i = \overline{z}_p^i(k)} , \tag{7.10}$$

$$\sigma_p^i(k+1) = \sigma_p^i(k) - \alpha_{\sigma_p^i} \frac{\partial E_k}{\partial \sigma_p^i} |_{\sigma_p^i = \sigma_p^i(k)} . \tag{7.11}$$

Let

$$A_{ik} \triangleq \prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{z_p(k) - \overline{z}_p^i(k)}{\sigma_p^i(k)})^2] . \tag{7.12}$$

then

$$y(k) = \sum_{i=1}^{I} \theta_i(k) \overline{y}_i(k) = \sum_{i=1}^{I} \frac{A_{ik} \overline{y}_i(k)}{\sum_{i=1}^{I} A_{ik}} . \tag{7.13}$$

The gradient terms can be obtained as [177].

$$\frac{\partial E_k}{\partial \overline{y}_i} |_{\overline{y}_i = \overline{y}_i(k)} = [\, y(k) - q(k) \,] \cdot \frac{A_{ik}}{\sum_{i'=1}^{I} A_{i'k}} . \tag{7.14}$$

$$\frac{\partial E_k}{\partial \bar{\xi}_p^i}\Big|_{\bar{\xi}_p^i=\bar{\xi}_p^i(k)} = [\, y(k)-q(k)\,]\cdot\frac{A_{ik}}{\sum_{i'=1}^{I} A_{i'k}}\cdot[\bar{y}_i(k)-y(k)]\cdot\frac{x_p(k)-\bar{\xi}_p^i(k)}{(\sigma_p^i(k))^2} \quad (7.15)$$

$$\frac{\partial E_k}{\partial \sigma_p^i}\Big|_{\sigma_p^i=\sigma_p^i(k)} = [\, y(k)-q(k)\,]\cdot\frac{A_{ik}}{\sum_{i'=1}^{I} A_{i'k}}\cdot[\bar{y}_i(k)-y(k)]\cdot\frac{[\,x_p(k)-\bar{\xi}_p^i(k)\,]^2}{(\sigma_p^i(k))^3}\,. \quad (7.16)$$

Although powerful, the BP approach is also well known for its weaknesses of generally slow convergence rate and the possibility of being trapped in local minima.

Observing eqs.(7.1-7.3), it is clear that expression (7.1) is linear in parameters $\bar{y}_i$, $i = 1,\ldots,I$. Therefore, if the parameters $\bar{\xi}_p^i$ and σ_p^i are determined, the well known least-square estimation (LSE) technique can be applied to estimate parameters $\bar{y}_i$. The parameter vector, $\overline{\mathbf{Y}}$, obtained in this way is the global optimal point in the parameter space $\overline{\mathcal{Y}}$, $\overline{\mathbf{Y}} \in \overline{\mathcal{Y}} \subset R^I$ [7, Chap.3].

In the following sections, we first briefly review the LSE approach, and then present a new FLS training scheme which combines the LSE technique with BP approach.

7.2.2 Least-Squares Estimation

Let

$$[\mathbf{\Theta}]_k \triangleq \begin{bmatrix} \mathbf{\Theta}^T(1) \\ \vdots \\ \mathbf{\Theta}^T(k) \end{bmatrix}. \quad (7.17)$$

if $[\mathbf{\Theta}]_k^T[\mathbf{\Theta}]_k$ is nonsingular, and parameter vector, $\overline{\mathbf{Y}}(k)$, is given by

$$\overline{\mathbf{Y}}(k) \triangleq ([\mathbf{\Theta}]_k^T[\mathbf{\Theta}]_k)^{-1}[\mathbf{\Theta}]_k^T Q_k\,. \quad (7.18)$$

then the loss function of eq.(7.4) is minimized, and this minimum is unique [7, Chap.3].

In control applications, the observations, $(\mathbf{z}(k), q(k))$, are obtained sequentially in real time. It is then desirable to make the computations recursive to save computation time. That is, the computation should be arranged in such a way that the results obtained at the (m-1)th step can be used to obtain the results at the mth step. This problem has also been well developed in the literature, and the results are given as follows [7, Chap.3]:

Assume $[\mathbf{\Theta}]_k^T[\mathbf{\Theta}]_k$ is nonsingular for all $k \geq k_0$, given $\overline{\mathbf{Y}}(k_0)$ and $S(k_0)$, the least-squares estimate $\overline{\mathbf{Y}}(k)$ then satisfies the recursive equations.

$$\overline{\mathbf{Y}}(k+1) = \overline{\mathbf{Y}}_k + K(k+1)[\, q(k) - \mathbf{\Theta}^T(k+1)\overline{\mathbf{Y}}(k)\,]\,. \tag{7.19}$$

$$\begin{aligned} K(k+1) &= S(k+1)\mathbf{\Theta}(k+1) \\ &= S(k)\mathbf{\Theta}(k+1)[\, I + \mathbf{\Theta}^T(k+1)S(k)\mathbf{\Theta}(k+1)\,]^{-1}\,. \end{aligned} \tag{7.20}$$

$$\begin{aligned} S(k+1) &= S(k) - S(k)\mathbf{\Theta}(k+1)[\, I + \mathbf{\Theta}^T(k+1)S(k)\mathbf{\Theta}(k+1)\,]^{-1}\mathbf{\Theta}^T(k+1)S(k) \\ &= [\, I - K(k+1)\mathbf{\Theta}^T(k+1)\,]S(k)\,. \end{aligned} \tag{7.21}$$

To obtain an initial condition for $S(k)$, it is necessary to choose $k = k_0$ such that $[\mathbf{\Theta}]_{k_0}^T[\mathbf{\Theta}]_{k_0}$ is nonsingular. The initial conditions are then

$$\begin{cases} S(k_0) \triangleq ([\mathbf{\Theta}]_{k_0}^T[\mathbf{\Theta}]_{k_0})^{-1}\,, \\ \overline{\mathbf{Y}}(k_0) \triangleq S(k_0)[\mathbf{\Theta}]_{k_0}^T Q_{k_0}\,. \end{cases} \tag{7.22}$$

Or simply let $S(0) \triangleq S_0 I$, where S_0 is some large positive number [41, Chap.7].

7.2.3 LSE-BP Training of FLS

Via the LSE approach it is easy to obtain the optimal parameter vector, $\overline{\mathbf{Y}}$, which minimizes the loss function, J_N, in eq.(7.4). For eq.(7.18) or the recursive equations (7.19-7.22) to be valid, the matrix $[\mathbf{\Theta}]_k^T[\mathbf{\Theta}]_k$ must be nonsingular. In addition, the parameters Ξ_p^l and σ_p^l are still to be determined.

At this point the possibility of the combination of the LSE and the BP approaches which includes the advantages and avoids the weaknesses of both, which we term the LSE-BP approach, become apparent.

The LSE-BP approach works as follows.

(i) Design the initial fuzzy logic system. Determine the number of fuzzy rules, I, and assign initial values for parameters, $\bar{x}_p^i$, σ_p^i and $\bar{y}_i$. Expert knowledge can be used in this stage. If there is no expert knowledge available at all, the initial values may be assigned randomly. Also, determine the learning rates, $\alpha_{\bar{y}_i}$, $\alpha_{\bar{x}_p^i}$ and $\alpha_{\sigma_p^i}$, for BP training rules, eqs.(7.9-7.11).

(ii) Obtain input-output pairs of the plant and FLS, $(\mathbf{z}(k), q(k))$ and $(\mathbf{z}(k), y(k))$, respectively. Check the estimation error, and if satisfactory, terminate the training process, otherwise, go to the next step.

(iii) Check the matrix $[\mathbf{\Theta}]_k^T[\mathbf{\Theta}]_k$ for nonsingularity. If it is nonsingular, use the recursive LSE equations, (7.19-7.22), to obtain the optimal parameter vector, $\bar{\mathbf{Y}}(k+1)$, otherwise, go to the next step.

(iv) If the matrix $[\mathbf{\Theta}]_k^T[\mathbf{\Theta}]_k$ is singular, use BP equations, (7.9-7.16), to update parameters $\bar{y}_i$, $\bar{x}_p^i$ and σ_p^i, otherwise, only update parameters $\bar{x}_p^i$ and σ_p^i. Then go back to step (ii).

7.3 Numerical Examples

Consider a nonlinear system which was used in [177].

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + g[u(k)] . \tag{7.23}$$

where the unknown function, $g(u)$, has the form

$$g(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u) . \tag{7.24}$$

Our objective is to identify the unknown function $g(u)$ using the FLS and optimal training approaches presented in previous sections. The above dynamic model, eq.(7.24), will be treated strictly as a black box, and is unknown to the identifiers.

The FLS of eq.(7.1) is used as the identifier. Its input is u, and its output is the estimation of the function g, denoted $\hat{g}$, i.e.,

$$\hat{g} = \mathbf{\Theta}^T(u')\overline{\mathbf{Y}} \ . \tag{7.25}$$

and the estimated system output, $\hat{y}$, is defined as

$$\hat{y}(k+1) = 0.3\hat{y}(k) + 0.6\hat{y}(k-1) + \hat{g} \ . \tag{7.26}$$

The input universe of discourse, $\mathcal{U} \subset R$, is partitioned into 15 primary fuzzy sets, U^i, $i = 1, \ldots, 15$. Therefore,

$$\begin{cases} \mathbf{\Theta}(u') \triangleq \{\theta_1(u'), \ldots, \theta_{15}(u')\}^T \ . \\ \overline{\mathbf{Y}} \triangleq \{\overline{y}_1, \ldots, \overline{y}_{15}\}^T \ . \end{cases} \tag{7.27}$$

and

$$\theta_i(u') = \frac{\mu_{U^i}(u')}{\sum_{i=1}^{15} \mu_{U^i}(u')} \ . \tag{7.28}$$

where the membership functions are defined as

$$\begin{aligned} \mu_{U^1}(u') &\triangleq \begin{cases} 1 \ . & if \quad u' < \overline{u}^1 \ . \\ \exp[-\frac{1}{2}(\frac{u'-\overline{u}^1}{\sigma^1})^2] \ . & if \quad u' \geq \overline{u}^1 \ . \end{cases} \\ \mu_{U^i}(u') &\triangleq \exp[-\frac{1}{2}(\frac{u'-\overline{u}^i}{\sigma^i})^2] \ . \quad i = 2, \ldots, 14 \\ \mu_{U^{15}}(u') &\triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{u'-\overline{u}^{15}}{\sigma^{15}})^2] \ . & if \quad u' < \overline{u}^{15} \ . \\ 1 \ . & if \quad u' \geq \overline{u}^{15} \ . \end{cases} \end{aligned} \tag{7.29}$$

In the above equations, u' is the filtered value of u, which is defined as

$$u' \triangleq f(u) \triangleq u \cdot \tau \ . \tag{7.30}$$

Initially, the centers of the primary fuzzy sets are defined as

$$\{\bar{u}^1, \bar{u}^2, \ldots, \bar{u}^8, \ldots, \bar{u}^{14}, \bar{u}^{15}\}^T \triangleq \{-7, -6, \ldots, 0, \ldots, 6, 7\}^T . \tag{7.31}$$

The shape parameters, σ^i, $i = 1, \ldots, 15$, are assigned random numbers uniformly distributed in (0.1, 0.6), this being an arbitrary range balancing with against strength of the membership functions. All the learning rates, $\alpha_{\bar{y}^i}$, $\alpha_{\bar{u}^i}$, and α_{σ^i}, are set at 0.1.

For inputs, $u(k)$, $k = 1, 2, \ldots$, being random numbers uniformly distributed in (-1, 1), the training processes for the FLS identifiers are shown in fig.7.1, where fig.7.1(a-b) shows the training of the FLS identifiers using LSE-BP and BP approaches, respectively. All training ends at $k = 150$. Figure 7.1(c) shows the estimation errors of the FLS identifiers for both training approaches. The superiority of the LSE-BP approach over BP approach is very clear.

The initial and final values of the FLS parameters, $\bar{u}^i$, σ^i and $\bar{y}^i$, $i = 1, \ldots, 15$, obtained with both LSE-BP and BP approaches are tabulated in Table 7.1.

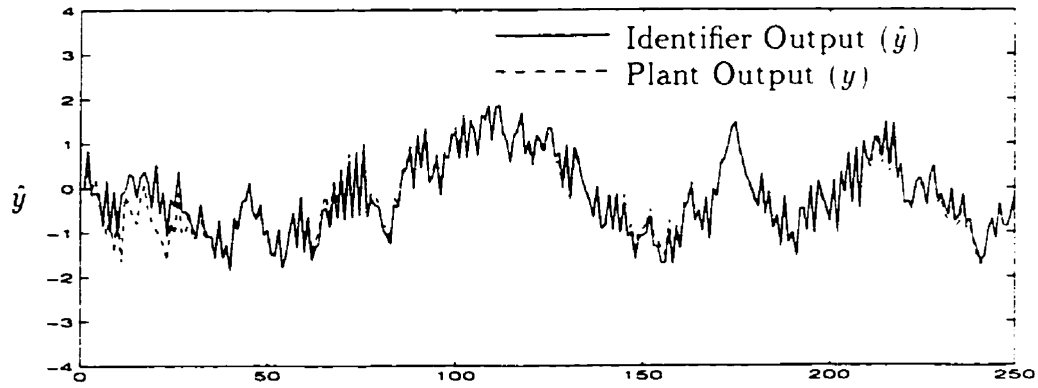
The trained FLS identifiers are tested by using them to predict the system outputs for the input

$$u(k) \triangleq \begin{cases} \sin(2\pi k/250) , & for\ 1 \le k \le 250\ and\ k \ge 501. \\ 0.5\sin(2\pi k/250) + 0.5\sin(2\pi k/25) , & for\ 251 \le k \le 500. \end{cases} \tag{7.32}$$

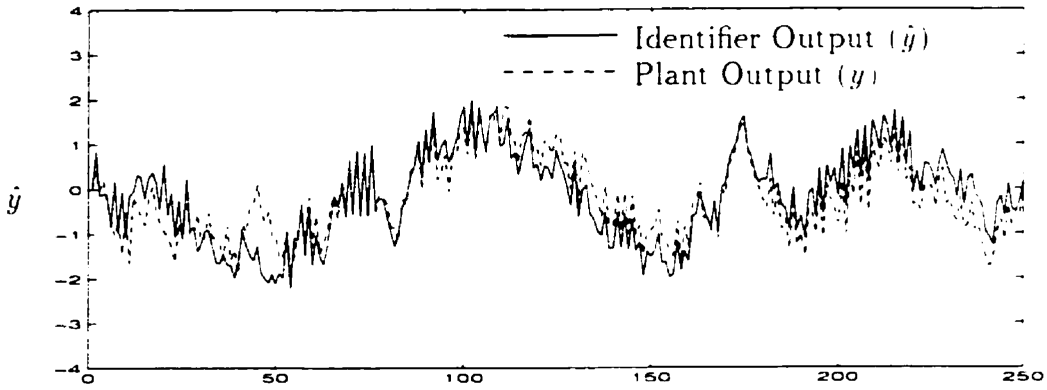
which is shown in fig.7.2(a). The outputs of the LSE-BP trained identifier and the BP trained identifier are illustrated in fig.7.2(b-c), respectively. For the same number of training steps, the performance of the LSE-BP trained identifier is much better than the other.

Table 7.1: Initial and Final Values of FLS parameters

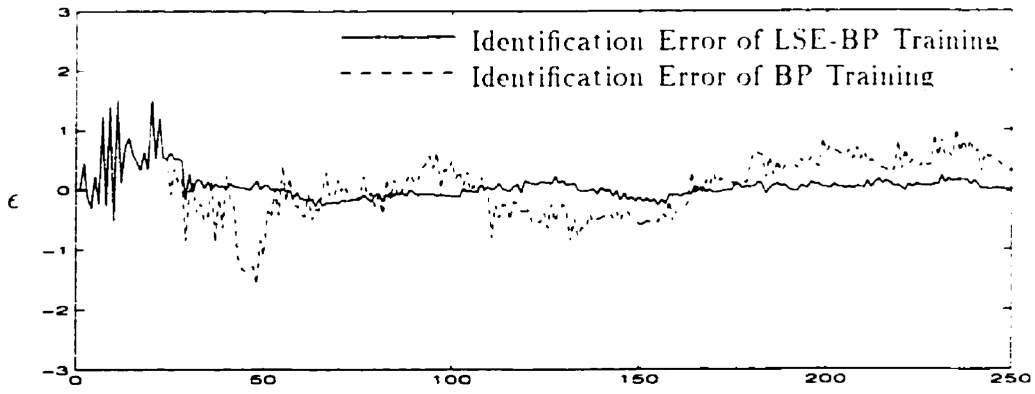
	$\bar{u}^i$			σ^i			$\bar{y}^i$		
i	Initial	LSE-BP	BP	Initial	LSE-BP	BP	Initial	LSE-BP	BP
1	-7	-6.915	-6.817	0.241	0.414	0.589	-0.992	0.023	-0.839
2	-6	-6.082	-6.215	0.383	0.564	0.494	0.732	-0.670	0.309
3	-5	-5.078	-5.165	0.200	0.418	0.505	-0.529	-0.471	-0.603
4	-4	-3.978	-4.001	0.157	0.080	0.031	-0.812	-0.391	-0.712
5	-3	-2.977	-2.921	0.551	0.653	0.761	-0.503	-0.389	-0.430
6	-2	-2.144	-2.206	0.525	0.223	0.003	0.723	-0.510	0.520
7	-1	-1.059	-1.076	0.405	0.543	0.654	-0.101	-0.626	-0.486
8	0	0.022	0.028	0.103	0.185	0.138	0.037	-0.136	0.002
9	1	1.070	1.066	0.514	0.682	0.596	0.190	0.599	0.390
10	2	2.037	2.054	0.212	0.246	0.182	-0.029	0.511	0.044
11	3	2.692	2.677	0.110	1.014	1.529	0.508	0.534	0.629
12	4	3.755	4.144	0.527	1.037	0.027	-0.989	0.250	-0.739
13	5	4.947	4.922	0.545	0.670	0.596	-0.479	0.603	-0.072
14	6	6.023	6.143	0.421	0.458	0.403	-0.667	0.757	-0.263
15	7	6.982	6.710	0.421	0.474	0.850	0.740	0.035	0.801



(a) LSE-BP Training



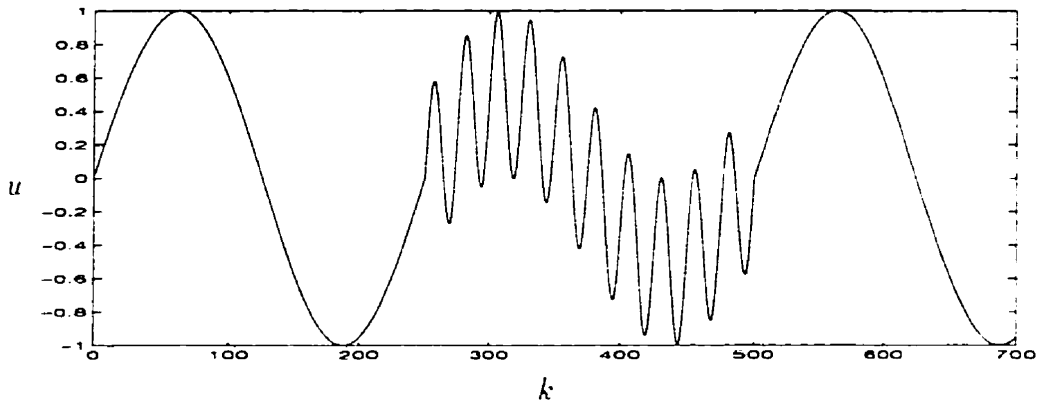
(b) BP Training



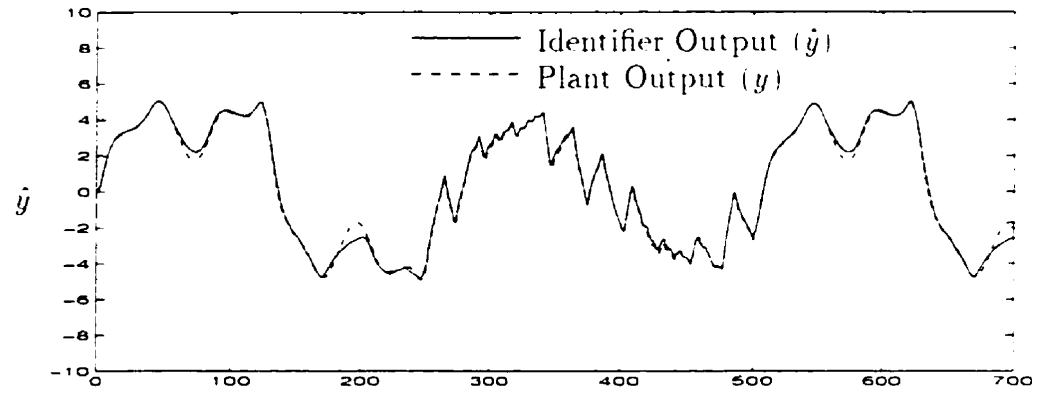
(c) Estimation Error

(Training Terminates at k=150)

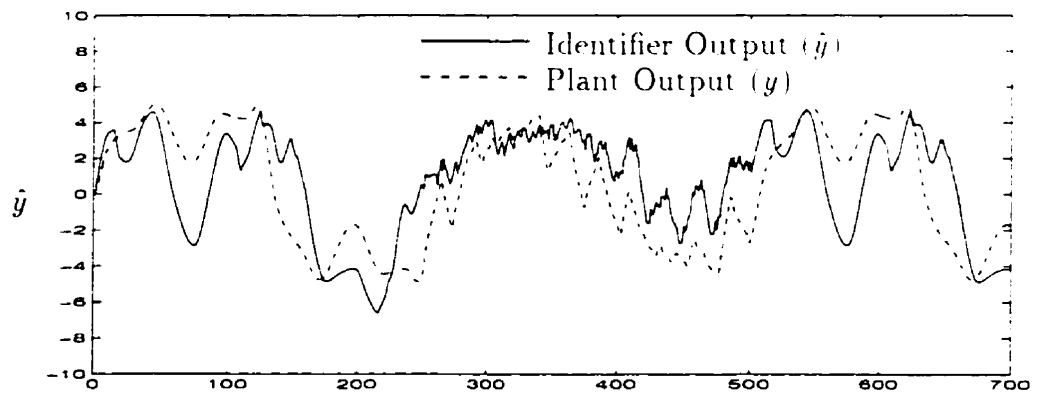
Figure 7.1: Training of the FLS Identifiers



(a) Input u



(b) LSE-BP Trained FLS



(c) BP Trained FLS

Figure 7.2: Testing of the Trained FLS Identifiers

7.4 Concluding Remarks

- A LSE-BP training scheme has been presented here, which is a combination of LSE approach and BP approach. The resulting optimal LSE-BP training scheme avoids the weakness of both the LSE and BP approaches, such as the rank requirement on the parameter matrix in LSE, and the slow convergence rate and the possibility of being trapped in local minima for the BP approach. Meanwhile, it combines the strengths of both LSE and BP, such as the wide applicability of BP and quick convergence rate and global optimality of LSE. Comparisons were made in simulation examples between the LSE-BP and BP approaches. The superiority of the LSE-BP approach over the BP approach is clearly demonstrated for this application.

- In eq.(7.17),

$$[\Theta]_N \triangleq \begin{bmatrix} \Theta^T(1) \\ \vdots \\ \Theta^T(N) \end{bmatrix}_{N \times I} . \tag{7.33}$$

  N is the size of data set and I is the size of fuzzy rule base. $[\Theta]_N$ is a N by I matrix. From matrix theory, it is known that a necessary condition for the matrix $[\Theta]_N^T[\Theta]_N$ to be nonsingular is that $N > I$ [66, App. A]. In control and identification applications, the number of fuzzy rules is usually much less than the number of data samples, and therefore, this condition is of no concern. If in certain situations in which the size of data set is small, for $[\Theta]_N^T[\Theta]_N$ to be nonsingular the maximal allowable number of fuzzy rules is $N - 1$, i.e., $I_{\max} = N - 1$.

- This author discovered an approach similar to the LSE-BP presented here in recently published literature [63], although an explicit comparison of the training process and the BP approach is not presented in this reference.

Chapter 8

The Dynamic Fuzzy Logic System and Nonlinear System Identification

8.1 Introduction

As indicated previously, fuzzy logic systems provide an effective framework for incorporating human linguistic descriptions into otherwise unknown systems, and their parameters have clear physical interpretations [88, 180, 206]. Impressive results have been achieved, both theoretically and practically, for identification and control of complicated systems, as outlined in chapter 2. However, the fuzzy logic systems used to date are static in nature. Since the physical systems of interest in identification and control applications are generally dynamic, this suggests that one might include dynamical elements into fuzzy logic systems, which would themselves then be more naturally integrated into dynamic systems to take advantage of these intrinsic dynamics. This would provide new tools in identification and control of dynamic systems.

With this in mind, it is proposed here to incorporate dynamics into fuzzy logic systems. The resulting fuzzy logic system, termed the Dynamic Fuzzy Logic System (DFLS), will be shown to possess the important universal approximation capability,

Further, based on the DFLS, a stable identification scheme, which is applicable to a large class of nonlinear dynamic systems, is developed via a Lyapunov synthesis approach.

To motivate the discussion and to serve as a vehicle for demonstrating application to nonlinear systems, the DFLS identification approach is applied to identify various nonlinear systems.

8.2 The DFLS and Universal Approximation

Consider a differential equation of the form

$$\dot{y} = -\alpha y + f_{FLS}(\mathbf{z}) \ . \tag{8.1}$$

where y is a scalar variable, α is a positive constant, and the second term on the right hand side of above equation, $f_{FLS}(\mathbf{z})$, represents the output of an ordinary fuzzy logic system, i.e., that shown in fig.4.1, and the vector, $\mathbf{z} = \{z_1, \ldots, z_P\}^T$, is the input of this fuzzy logic system. Equation (8.1) is illustrated in fig.8.1. We term this system the *dynamic fuzzy logic system*, abbreviated as DFLS, for its explicit dependence on time.

In the rest of this work, we restrict the function $f_{FLS}(\mathbf{z})$, i.e., the ordinary fuzzy logic system or fuzzy logic system in short, to be of the form of eq.(4.45), i.e.,

$$f_{FLS}(\mathbf{z}) \triangleq \boldsymbol{\Theta}^T(\mathbf{z})\overline{\mathbf{Y}} \ . \tag{8.2}$$

where $\boldsymbol{\Theta}(\mathbf{z})$ and $\overline{\mathbf{Y}}$ are defined in eq.(4.44),

$$\begin{cases} \boldsymbol{\Theta}(\mathbf{z}) \triangleq \{\theta_1(\mathbf{z}), \ldots, \theta_I(\mathbf{z})\}^T \ , \\ \overline{\mathbf{Y}} \triangleq \{\overline{y}_1, \ldots, \overline{y}_I\}^T \ , \end{cases} \tag{8.3}$$

and $\theta_i(\mathbf{z})$ is given by eq.(4.43) as,

$$\theta_i(\mathbf{z}) \triangleq \frac{\prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{z_p - \overline{z}_p^i}{\sigma_p^i})^2]}{\sum_{i=1}^{I} \prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{z_p - \overline{z}_p^i}{\sigma_p^i})^2]} \ . \tag{8.4}$$

Equations (8.2–8.4) represent a static fuzzy logic system characterized by a singleton fuzzifier, CM defuzzifier, the product implication rule, the sup-product compositional operator, Gaussian type membership functions for primary fuzzy sets, and the "$\star$" operators representing the algebraic product operation. This specific static FLS has been theoretically justified as a universal approximator [184].

Next, substituting eq.(8.2) into (8.1) yields

$$\dot{y} = -\alpha y + \mathbf{\Theta}^T(\mathbf{z})\overline{\mathbf{Y}} \ . \tag{8.5}$$

We shall now show that the DFLS of eq.(8.5) retains this capability of uniformly approximating a large class of nonlinear functions to any desired degree of accuracy.

Consider a nonlinear system described by differential equation in the form

$$\dot{x} = f(\mathbf{z}) \ . \tag{8.6}$$

where x is any scalar element of the vector $\mathbf{z} \in \mathcal{Z} \subset R^P$, i.e., $x \in \{z_1, z_2, \ldots, z_P\}$. $\mathbf{z}$ may be composed of both system states and external inputs to the system. $f : R^P \rightarrow R$ is a continuous static nonlinear map defined on a compact set $\mathcal{Z}$.

Theorem 8.1 *For any $\varepsilon > 0$, there exists a DFLS in the form of eq.(8.5) with $\mathbf{z}$, as described above, defined on a compact set $\mathcal{Z} \subset R^P$, such that*

$$\sup_{z \in \mathcal{Z}} \mid y - x \mid < \varepsilon \ . \tag{8.7}$$

Proof:

Rewrite eq.(8.6) as

$$\dot{x} = -\alpha x + g(\mathbf{z}) \ . \tag{8.8}$$

where

$$g(\mathbf{z}) \triangleq \alpha x + f(\mathbf{z}) \ . \tag{8.9}$$

Subtracting eq.(8.8) from eq.(8.5) gives

$$\dot{\epsilon} = -\alpha\epsilon + [\mathbf{\Theta}^T(\mathbf{z})\overline{\mathbf{Y}} - g(\mathbf{z})] \ . \tag{8.10}$$

where

$$\epsilon \triangleq y - x \ . \tag{8.11}$$

The solution to eq.(8.10) is

$$\epsilon(t) = [\exp(-\alpha t)]\epsilon(0) + \int_0^t [\exp(-\alpha(t-\tau))][\mathbf{\Theta}^T(\mathbf{z})\overline{\mathbf{Y}} - g(\mathbf{z})]d\tau \ . \tag{8.12}$$

Or

$$| \epsilon(t) | \leq | [\exp(-\alpha t)]\epsilon(0) | + \int_0^t [\exp(-\alpha(t-\tau))] | \mathbf{\Theta}^T(\mathbf{z})\overline{\mathbf{Y}} - g(\mathbf{z}) | d\tau \ . \tag{8.13}$$

It is straight forward to verify that, if $\epsilon(0) \neq 0$, $\forall\ \delta_0 > 0$ and

$$t_0(\delta_0) \in \{t : \ t > \max\{0, \ \frac{1}{\alpha}\ln[\frac{|\epsilon(0)|}{\delta_0}]\}\} \ . \tag{8.14}$$

then

$$| [\exp(-\alpha t_0)]\epsilon(0) | < \delta_0 \ . \tag{8.15}$$

From the universal approximation theorem for fuzzy logic systems [177], we have that for compact set $\mathcal{Z} \subset R^P$ and $\forall\ \delta > 0$, $\exists\ \mathbf{\Theta}$ and $\overline{\mathbf{Y}}$, such that

$$\sup_{\mathbf{z}\in\mathcal{Z}} | \mathbf{\Theta}^T(\mathbf{z})\overline{\mathbf{Y}} - g(\mathbf{z}) | < \delta \ . \tag{8.16}$$

Thus

$$\begin{aligned} \int_0^t [\exp(-\alpha(t-\tau))] | \mathbf{\Theta}^T(\mathbf{z})\overline{\mathbf{Y}} - g(\mathbf{z}) | d\tau &< \int_0^t \delta[\exp(-\alpha(t-\tau))]d\tau \\ &= \frac{\delta}{\alpha}[1 - \exp(-\alpha t)] \\ &< \frac{\delta}{\alpha} \ . \end{aligned} \tag{8.17}$$

Using eqs. (8.15) and (8.17) in eq.(8.13) yields

$$| \epsilon(t) | < \delta_0 + \frac{\delta}{\alpha} . \tag{8.18}$$

Now let

$$\varepsilon \triangleq \delta_0 + \frac{\delta}{\alpha} , \tag{8.19}$$

then

$$| y - x | < \varepsilon . \tag{8.20}$$

This completes the proof.

Although this theorem reveals the universal approximation property of a DFLS, it only indicates that a DFLS can be a universal approximator, without providing any guidance on how one might construct such a DFLS to approximate a given physical system. In the following, we will address this latter point by developing a stable identification algorithm that is based on the DFLS, and is applicable to large class of nonlinear systems.

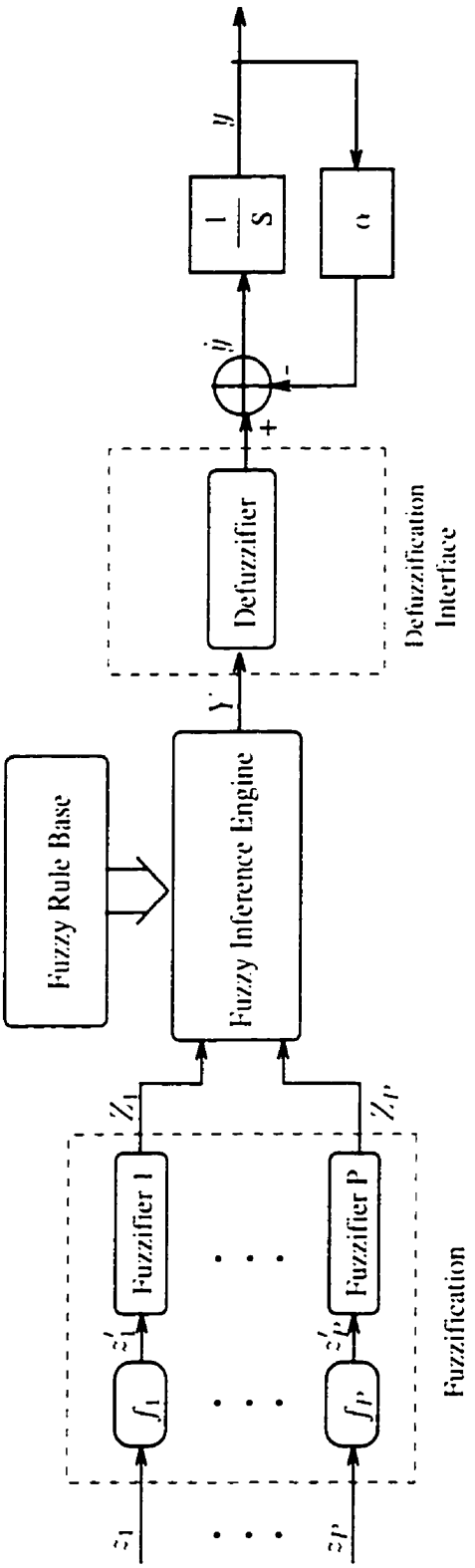


Figure 8.1: Dynamic Fuzzy Logic System

8.3 DFLS Based Identification for Nonlinear Dynamic Systems and Stability Analysis

8.3.1 Preliminary

In this section, we develop a DFLS based stable identification algorithm for identification of nonlinear dynamic systems. Consider a general dynamic system of the form

$$\dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{u}) \,. \tag{8.21}$$

where $\mathbf{x} \triangleq \{x_1, \ldots, x_N\}^T \in R^N$ and $\mathbf{u} \triangleq \{u_1, \ldots, u_M\}^T \in R^M$ are the state and external input vectors of the physical process, respectively. N and M represent the total number of states and external inputs. $F : R^{N+M} \rightarrow R^N$ is continuous nonlinear vector function defined on a compact set $\mathcal{Z} \subset R^{N+M}$.

For ease of later discussion, we rewrite eq.(8.21) as a set of state equations, i.e., for $k = 1, \ldots, N$,

$$\dot{x}_k = f_k(\mathbf{x}, \mathbf{u}) \,. \tag{8.22}$$

where $f_k : R^{N+M} \rightarrow R$ is continuous nonlinear function defined on the compact set $\mathcal{Z}$. A slight difference in notation between eqs.(8.22) and (8.6) is apparent. In eq.(8.6), the vector $\mathbf{z}$, composed of both states and external inputs of the physical process, is explicitly decomposed in eq.(8.22) into $\mathbf{x}$ and $\mathbf{u}$, i.e., $\mathbf{z} = \{\mathbf{x}, \mathbf{u}\}^T$, $P = N + M$. Here, we impose a mild restriction on this system:

Assumption 8.1 *For given admissible input* $\mathbf{u}$ *and any finite initial condition* $\mathbf{x}(0)$, *the state,* $\mathbf{x}(t) = \{x_1(t), \ldots, x_N(t)\}^T$, *is bounded.*

The DFLS of eq.(8.5) is apparently characterized by the free design parameters ξ_{j_p}, σ_{j_p}, $\overline{y}_i$, and α, where $i = 1, \ldots, I$, $p = 1, \ldots, P$, and $j_p = 1, \ldots, J_p$. J_p is the

number of primary fuzzy sets in the universe of discourse $\mathcal{Z}_p$, which is also a design parameter.

In the rest of this work, we assume that for $p = 1, \ldots, P$, the fuzzy partition parameter J_p, the membership function parameters $\bar{z}_{j_p}$ and σ_{j_p}, as well as the parameter α are designed off-line, leaving only $\overline{\mathbf{Y}} = \{\bar{y}_1, \ldots, \bar{y}_I\}^T$ as an on-line adjustable parameter vector. The off-line design process is the same as that for static fuzzy logic systems, such as that of chapter 6, and this is the design stage where engineering knowledge and intuition about the system concerned can be incorporated.

Our objective here is to develop a stable training algorithm for the DFLS such that it can identify an unknown nonlinear dynamic system in the form of eq.(8.22), with the identification error being bounded and as small as possible, ideally converging to zero, and at the same time, the parameters themselves should also be bounded.

8.3.2 DFLS Identification Algorithm

To identify the kth state of eq.(8.22), x_k, $k \in \{1, \ldots, N\}$, a DFLS of the following form is used:

$$\dot{y}_k = -\alpha_k y_k + \boldsymbol{\Theta}_k^T(\mathbf{x}, \mathbf{u})\overline{\mathbf{Y}}_k , \qquad \alpha_k > 0 . \tag{8.23}$$

where $\mathbf{x}$ and $\mathbf{u}$ are the states and external inputs of the physical process, $\boldsymbol{\Theta}_k \triangleq \{\theta_{k1}, \ldots, \theta_{kI}\}^T$, $\overline{\mathbf{Y}}_k \triangleq \{\bar{y}_{k1}, \ldots, \bar{y}_{kI}\}^T$. The subscript k indicates that the identifier is for the kth state. This formulation implicitly assumes that all the states are available.

Since we may be interested in identifying only certain states, the subscript k need not include all states from 1 to N. In addition, it may be unnecessary to feed all the states, $\{x_1, \ldots, x_N\}^T$, and external inputs, $\{u_1, \ldots, u_M\}^T$, into the fuzzy logic system, since a specific state may have little correlation with certain other states

and external inputs, which may be determined from physical intuition. For example, in the case of a flexible link manipulator, the beam vibration might bear little direct relationship to the angular position of the motor hub. In any case, eq.(8.23) is sufficiently general to cover these situations.

Equation (8.22) can be rewritten as

$$\dot{x}_k = -a_k x_k + a_k x_k + f_k(\mathbf{x}, \mathbf{u}) \,, \quad k \in \{1, \ldots, N\}. \tag{8.24}$$

By denoting the identification error of the kth state as

$$\epsilon_k \stackrel{\Delta}{=} y_k - x_k \,, \tag{8.25}$$

and subtracting eq.(8.24) from eq.(8.23), we get

$$\dot{\epsilon}_k = -a_k \epsilon_k + [\mathbf{\Theta}_k^T(\mathbf{x}, \mathbf{u})\overline{\mathbf{Y}}_k - a_k x_k - f_k(\mathbf{x}, \mathbf{u})] \,. \tag{8.26}$$

where $\overline{\mathbf{Y}}_k$ is the adjustable parameter vector. Define

$$r_k(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}_k) \stackrel{\Delta}{=} \mathbf{\Theta}_k^T(\mathbf{x}, \mathbf{u})\overline{\mathbf{Y}}_k - a_k x_k - f_k(\mathbf{x}, \mathbf{u}) \,. \tag{8.27}$$

which can be viewed as the static modeling error of the DFLS. Equations (8.26) and (8.27) together give

$$\dot{\epsilon}_k = -a_k \epsilon_k + r_k(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}_k) \,. \tag{8.28}$$

The identification error, ϵ_k, is evidently controlled by $r_k(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}_k)$. We shall show below that under certain conditions, the identification error can be made arbitrarily small. We first have the following preliminary result:

Lemma 8.1 *For given* $0 < M_{r_k} < \infty$, $\exists\, \overline{\mathbf{Y}}'_k \in \{\overline{\mathbf{Y}}_k : \| \overline{\mathbf{Y}}_k \| \leq M_{\overline{\mathbf{Y}}_k}\}$.
such that

$$\sup_{\{\mathbf{x},\mathbf{u}\}\in\mathcal{Z}} | r_k(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}'_k) | \leq M_{r_k}. \qquad k \in \{1, \ldots, N\} \,. \tag{8.29}$$

where M_{r_k} *and* $M_{\overline{\mathbf{Y}}_k}$ *are positive design constants.*

Proof : This follows directly from the universal approximation theorem of the static FLS[177].

Remarks:

- In Lemma 8.1, we may further denote $\overline{\mathbf{Y}}_k^*$ as the optimal parameter vector which minimizes the static modeling error, $r_k(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}_k')$, for $(\mathbf{x}, \mathbf{u}) \in \Xi \subset R^{N+M}$, i.e.

$$\overline{\mathbf{Y}}_k^* \triangleq arg \min_{\{\overline{\mathbf{Y}}_k'\}} \{ \sup_{\{\mathbf{x},\mathbf{u}\} \in \Xi} | r_k(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}_k') | \} . \tag{8.30}$$

- Lemma 8.1 indicates only the existence of such a parameter vector, but not how to find it. It will be shown that by adjusting $\overline{\mathbf{Y}}_k$ with the training law presented below, it is possible to find a $\overline{\mathbf{Y}}_k$ which can approximate $\overline{\mathbf{Y}}_k^*$ satisfactorily.

We can rewrite eq.(8.22) as

$$\dot{x}_k = -\alpha_k x_k - \mathbf{\Theta}_k^T(\mathbf{x}, \mathbf{u}) \overline{\mathbf{Y}}_k^* - r_k(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}_k^*) , \quad k \in \{1, \ldots, N\}. \tag{8.31}$$

and subtract eq.(8.31) from eq.(8.23) to give

$$\dot{\epsilon}_k = -\alpha_k \epsilon_k - \mathbf{\Theta}_k^T(\mathbf{x}, \mathbf{u}) \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} + r_k(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}_k^*) , \quad k \in \{1, \ldots, N\}. \tag{8.32}$$

where

$$\mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \triangleq \overline{\mathbf{Y}}_k - \overline{\mathbf{Y}}_k^* . \tag{8.33}$$

Following the well known Lyapunov synthesis approach, we now obtain a training law for $\overline{\mathbf{Y}}_k$ such that the identification error, ϵ_k, is bounded, and even asymptotically approaches zero under certain conditions. To avoid circumstances under which the parameter vector $\overline{\mathbf{Y}}_k$ becomes too large or even drifts to infinity[133], a projection algorithm modification, which has been successfully implemented by various researchers[133, 42, 57], is used in the training process to guarantee the boundedness

of the parameters. The main results are summarized in the following theorem. We first specify the training law to be

$$\dot{\overline{\mathbf{Y}}}_k = -\mathbf{H}_k \mathbf{\Theta}_k(\mathbf{x}, \mathbf{u}) \epsilon_k - S_k \beta_k \mathbf{H}_k \overline{\mathbf{Y}}_k \ . \quad k \in \{1, \ldots, N\} \ . \tag{8.34}$$

where $\mathbf{H}_k$ is a constant positive definite symmetric matrix. S_k is a switch defined as

$$S_k = \begin{cases} 0 \ . \ \ if \ \ \| \ \overline{\mathbf{Y}}_k \ \| < M_{\overline{\mathbf{Y}}_k} \ . \\ \qquad or \ \ \| \ \overline{\mathbf{Y}}_k \ \| = M_{\overline{\mathbf{Y}}_k} \ \ and \ \ \mathbf{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k \epsilon_k \geq 0 \ . \\ \qquad or \ \ \| \ \overline{\mathbf{Y}}_k \ \| > M_{\overline{\mathbf{Y}}_k} \ \ and \ \ \mathbf{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k \epsilon_k > 0 \ . \\ \\ 1 \ . \ \ otherwise. \end{cases} \tag{8.35}$$

and β_k is a positive design parameter which satisfies

$$\beta_k \begin{cases} \geq -\frac{\mathbf{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k}{\overline{\mathbf{Y}}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k} \epsilon_k \ . \ \ if \ \ \| \ \overline{\mathbf{Y}}_k \ \| = M_{\overline{\mathbf{Y}}_k} \ \ and \ \ \mathbf{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k \epsilon_k < 0 \ . \\ \\ > -\frac{\mathbf{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k}{\overline{\mathbf{Y}}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k} \epsilon_k \ . \ \ if \ \ \| \ \overline{\mathbf{Y}}_k \ \| > M_{\overline{\mathbf{Y}}_k} \ \ and \ \ \mathbf{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k \epsilon_k \leq 0 \ . \end{cases} \tag{8.36}$$

Theorem 8.2 *Consider an unknown nonlinear dynamic system in the form of eq.(8.22), which is to be identified by the DFLS of eq.(8.23), by adjusting the parameter vector* $\overline{\mathbf{Y}}_k$ *of the DFLS with the training law of eq.(8.34). The DFLS identifier possesses following properties.*

8.2.1 $\| \ \overline{\mathbf{Y}}_k \ \| \leq M_{\overline{\mathbf{Y}}_k}$;

8.2.2 $| \ \epsilon_k \ | \leq M_{\epsilon_k}$, *where* M_{ϵ_k} *is a positive constant;*

8.2.3 $\int_0^t \epsilon_k^2(\tau) d\tau \leq a + b \int_0^t r_k^2(\tau) d\tau$, *where* a *and* b *are positive constants, and if* $r_k(t) \in \mathcal{L}_2[0, \infty)$, *i.e.*, $(\int_0^\infty \| \ r_k(\tau) \ \|^2 \ d\tau)^{\frac{1}{2}} < \infty$, *then,*

$\lim_{t \to \infty} | \ \epsilon_k(t) \ | = 0$.

8.3.3 Proof of Theorem 8.2

Proof of 8.2.1

Consider the Lyapunov function candidate.

$$V_{\overline{\mathbf{Y}}_k} = \frac{1}{2}\overline{\mathbf{Y}}_k^T\overline{\mathbf{Y}}_k \ . \tag{8.37}$$

then.

$$\dot{V}_{\overline{\mathbf{Y}}_k} = \dot{\overline{\mathbf{Y}}}_k^T\overline{\mathbf{Y}}_k \ ; \tag{8.38}$$

- If $\| \overline{\mathbf{Y}}_k \| < M_{\overline{\mathbf{Y}}_k}$, whether $\dot{V}_{\overline{\mathbf{Y}}_k}$ is positive or negative is of no concern.

- If $\| \overline{\mathbf{Y}}_k \| = M_{\overline{\mathbf{Y}}_k}$ and $\boldsymbol{\Theta}_k^T\mathbf{H}_k\overline{\mathbf{Y}}_k\epsilon_k \geq 0$, use of eqs.(8.34-8.35) in eq.(8.38) yields

$$\dot{V}_{\overline{\mathbf{Y}}_k} = -\boldsymbol{\Theta}_k^T\mathbf{H}_k\overline{\mathbf{Y}}_k\epsilon_k \leq 0 \tag{8.39}$$

  Therefore, $\| \overline{\mathbf{Y}}_k \|$ is bounded.

- If $\| \overline{\mathbf{Y}}_k \| = M_{\overline{\mathbf{Y}}_k}$ and $\boldsymbol{\Theta}_k^T\mathbf{H}_k\overline{\mathbf{Y}}_k\epsilon_k < 0$, use of eqs.(8.34-8.36) in eq.(8.38) results in

$$\dot{V}_{\overline{\mathbf{Y}}_k} = -\boldsymbol{\Theta}_k^T\mathbf{H}_k\overline{\mathbf{Y}}_k\epsilon_k - \beta_k\overline{\mathbf{Y}}_k^T\mathbf{H}_k\overline{\mathbf{Y}}_k \leq 0 \tag{8.40}$$

  Again, $\| \overline{\mathbf{Y}}_k \|$ is bounded.

- If $\| \overline{\mathbf{Y}}_k \| > M_{\overline{\mathbf{Y}}_k}$, i.e., $\| \overline{\mathbf{Y}}_k \|$ exceeds its *nominal bound*, $M_{\overline{\mathbf{Y}}_k}$, which may occur during the training process, and as the training is implemented computationally as

$$\overline{\mathbf{Y}}_k(t + dt) = \overline{\mathbf{Y}}_k(t) + \dot{\overline{\mathbf{Y}}}_k(t)dt \ . \tag{8.41}$$

  (where $\dot{\overline{\mathbf{Y}}}_k(t)$ is given by the training law eq.(8.34)), then in the situation in which $\| \overline{\mathbf{Y}}_k(t) \|$ is very close to the nominal bound, $M_{\overline{\mathbf{Y}}_k}$, but still within

this bound, with a gain $\mathbf{H}_k$ and the time increment "dt" being *sufficiently large*, the resulting $\| \overline{\mathbf{Y}}_k(t+dt) \|$ can exceed the prespecified nominal bound $M_{\overline{\mathbf{Y}}_k}$ by the norm of the training increment $\dot{\overline{\mathbf{Y}}}_k(t)dt$, at most. Now, use of eqs.(8.34–8.36) in eq.(8.38) yields

$$\begin{cases} For \quad \boldsymbol{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k \epsilon_k > 0 \quad : \quad \dot{V}_{\overline{\mathbf{Y}}_k} = -\boldsymbol{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k \epsilon_k < 0 \ . \\ For \quad \boldsymbol{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k \epsilon_k \leq 0 \quad : \quad \dot{V}_{\overline{\mathbf{Y}}_k} = -\boldsymbol{\Theta}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k \epsilon_k - \beta_k \overline{\mathbf{Y}}_k^T \mathbf{H}_k \overline{\mathbf{Y}}_k < 0 \ . \end{cases} \tag{8.42}$$

which indicates that as soon as $\| \overline{\mathbf{Y}}_k \|$ exceeds its nominal bound, $M_{\overline{\mathbf{Y}}_k}$, it immediately returns within this bound. Therefore, this excess is *small* and *temporary*. Formally, for the compact set Ξ, and any given $\mathbf{H}_k$ and dt,

$$\delta M_{\overline{\mathbf{Y}}_k} \triangleq \sup_{(\mathbf{x},\mathbf{u}) \in \Xi} \| \dot{\overline{\mathbf{Y}}}_k(t) dt \| \ , \tag{8.43}$$

then we have

$$\| \overline{\mathbf{Y}}_k(t+dt) \| \leq M_{\overline{\mathbf{Y}}_k} + \delta M_{\overline{\mathbf{Y}}_k} \ . \tag{8.44}$$

which indicates the boundedness of $\overline{\mathbf{Y}}_k$.

This concludes the proof of 8.2.1.

Proof of 8.2.2

Consider the Lyapunov function candidate

$$V = \frac{1}{2}[\epsilon_k^2 + \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k}^T \mathbf{H}_k^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k}] \ . \tag{8.45}$$

Its derivative is

$$\dot{V} = \epsilon_k \dot{\epsilon}_k + \dot{\boldsymbol{\Delta}}_{\overline{\mathbf{Y}}_k}^T \mathbf{H}_k^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} = \epsilon_k \dot{\epsilon}_k + \dot{\overline{\mathbf{Y}}}_k^T \mathbf{H}_k^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \ . \tag{8.46}$$

Substituting eq.(8.32) into eq.(8.46) yields

$$\dot{V} = -\alpha_k \epsilon_k^2 + \boldsymbol{\Theta}_k^T(\mathbf{x},\mathbf{u}) \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \epsilon_k + r_k(\mathbf{x},\mathbf{u},\boldsymbol{\Theta}_k,\overline{\mathbf{Y}}_k^*)\epsilon_k + \dot{\overline{\mathbf{Y}}}_k^T \mathbf{H}_k^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \ . \tag{8.47}$$

and use of eq.(8.34) in eq.(8.47) yields

$$\begin{aligned} \dot{V} &= -\alpha_k \epsilon_k^2 + \mathbf{\Theta}_k^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \epsilon_k + r_k \epsilon_k + (-\mathbf{\Theta}_k^T \mathbf{H}_k \epsilon_k - S_k \beta_k \overline{\mathbf{Y}}_k^T \mathbf{H}_k) \mathbf{H}_k^{-1} \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \\ &= -\alpha_k \epsilon_k^2 + r_k \epsilon_k - S_k \beta_k \overline{\mathbf{Y}}_k^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} . \end{aligned} \tag{8.48}$$

where all arguments have been suppressed. From (8.35), only for the case of $\| \overline{\mathbf{Y}}_k \| \geq M_{\overline{\mathbf{Y}}_k}$ with $S_k = 1$ need the last term of eq.(8.48) be considered. Now

$$\begin{aligned} \overline{\mathbf{Y}}_k^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} &= (\overline{\mathbf{Y}}_k - \overline{\mathbf{Y}}_k^* + \overline{\mathbf{Y}}_k^*)^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} = \mathbf{\Delta}_{\overline{\mathbf{Y}}_k}^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} + \overline{\mathbf{Y}}_k^{*T} \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \\ &= \frac{1}{2} \mathbf{\Delta}_{\overline{\mathbf{Y}}_k}^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} + \frac{1}{2} (\mathbf{\Delta}_{\overline{\mathbf{Y}}_k}^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} + 2 \overline{\mathbf{Y}}_k^{*T} \mathbf{\Delta}_{\overline{\mathbf{Y}}_k}) \\ &= \frac{1}{2} \mathbf{\Delta}_{\overline{\mathbf{Y}}_k}^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} + \frac{1}{2} (\overline{\mathbf{Y}}_k^T + \overline{\mathbf{Y}}_k^{*T}) \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \\ &= \frac{1}{2} \mathbf{\Delta}_{\overline{\mathbf{Y}}_k}^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} + \frac{1}{2} \overline{\mathbf{Y}}_k^T \overline{\mathbf{Y}}_k - \frac{1}{2} \overline{\mathbf{Y}}_k^{*T} \overline{\mathbf{Y}}_k^* \\ &= \frac{1}{2} \| \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 + \frac{1}{2} \| \overline{\mathbf{Y}}_k \|^2 - \frac{1}{2} \| \overline{\mathbf{Y}}_k^* \|^2 . \end{aligned} \tag{8.49}$$

where $\| \cdot \|$ is the Euclidean vector norm. From Lemma 8.1, $\| \overline{\mathbf{Y}}_k^* \| \leq M_{\overline{\mathbf{Y}}_k}$, and noting that $\| \overline{\mathbf{Y}}_k \| \geq M_{\overline{\mathbf{Y}}_k}$, shows that the last term of eq.(8.48) satisfies,

$$S_k \beta_k \overline{\mathbf{Y}}_k^T \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \geq \frac{1}{2} S_k \beta_k \| \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 \geq 0 . \tag{8.50}$$

Use of eq.(8.50) in eq.(8.48) now yields

$$\begin{aligned} \dot{V} &\leq -\alpha_k \epsilon_k^2 - \frac{1}{2} S_k \beta_k \| \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 + r_k \epsilon_k \\ &= -\frac{1}{2} \alpha_k \epsilon_k^2 - \frac{1}{2} \beta_k \| \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 + (1 - S_k) \frac{1}{2} \beta_k \| \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 \\ &\quad - \frac{1}{2} \alpha_k (\epsilon_k^2 - 2 \frac{r_k}{\alpha_k} \epsilon_k) . \end{aligned} \tag{8.51}$$

furthermore, again with the help of Lemma 8.1,

$$-\frac{1}{2} \alpha_k (\epsilon_k^2 - 2 \frac{r_k}{\alpha_k} \epsilon_k) = -\frac{1}{2} \alpha_k (\epsilon_k - \frac{r_k}{\alpha_k})^2 + \frac{r_k^2}{2\alpha_k} \leq \frac{r_k^2}{2\alpha_k} \leq \frac{(M_{r_k})^2}{2\alpha_k} . \tag{8.52}$$

and

$$\begin{aligned} (1 - S_k) \frac{1}{2} \beta_k \| \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 &\leq \frac{1}{2} \beta_k \| \mathbf{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 \leq \frac{1}{2} \beta_k (\| \overline{\mathbf{Y}}_k \| + \| \overline{\mathbf{Y}}_k^* \|)^2 \\ &\leq 2 \beta_k (M_{\overline{\mathbf{Y}}_k})^2 . \end{aligned} \tag{8.53}$$

and

$$-\frac{1}{2}\beta_k \| \Delta_{\overline{\mathbf{Y}}_k} \|^2 \leq -\frac{1}{2}\beta_k \frac{\Delta_{\overline{\mathbf{Y}}_k}^T \mathbf{H}_k^{-1} \Delta_{\overline{\mathbf{Y}}_k}}{\lambda_{\max}(\mathbf{H}_k^{-1})} . \tag{8.54}$$

where $\lambda_{\max}(\mathbf{H}_k^{-1})$ is the maximum eigenvalue of $\mathbf{H}_k^{-1}$. Define

$$\kappa \triangleq \min\{ \alpha_k, \frac{\beta_k}{\lambda_{\max}(\mathbf{H}_k^{-1})} \} . \tag{8.55}$$

Now using eqs.(8.52) to (8.55) in eq.(8.51) yields

$$\begin{aligned} \dot{V} &\leq -\kappa(\frac{1}{2}\epsilon_k^2 + \frac{1}{2}\Delta_{\overline{\mathbf{Y}}_k}^T \mathbf{H}_k^{-1} \Delta_{\overline{\mathbf{Y}}_k}) + 2\beta_k (M_{\overline{\mathbf{Y}}_k})^2 + \frac{(M_{r_k})^2}{2\alpha_k} \\ &\leq -\kappa V + C . \end{aligned} \tag{8.56}$$

where

$$C \triangleq 2\beta_k (M_{\overline{\mathbf{Y}}_k})^2 + \frac{(M_{r_k})^2}{2\alpha_k} . \tag{8.57}$$

which is a constant. That is

$$\dot{V} \leq -\kappa V + C . \tag{8.58}$$

Therefore,

$$\dot{V} \leq 0 , \quad if \quad V \geq \frac{C}{\kappa} . \tag{8.59}$$

Thus, if $V < \frac{C}{\kappa}$, then V is bounded, implying boundedness of ϵ_k and $\Delta_{\overline{\mathbf{Y}}_k}$. If $V \geq \frac{C}{\kappa}$, then eq.(8.59) indicates the boundedness of V, again implying the boundedness of ϵ_k and $\Delta_{\overline{\mathbf{Y}}_k}$. Therefore, there exists a positive constant, M_{ϵ_k}, such that $|\epsilon_k| \leq M_{\epsilon_k}$. This completes the proof of 8.2.2.

Proof of 8.2.3

From eqs.(8.51) and (8.52), we have

$$\begin{aligned} \dot{V} &\leq -\alpha_k \epsilon_k^2 - \frac{1}{2} S_k \beta_k \| \Delta_{\overline{\mathbf{Y}}_k} \|^2 + r_k \epsilon_k \leq -\alpha_k \epsilon_k^2 + r_k \epsilon_k \\ &\leq -\frac{1}{2}\alpha_k \epsilon_k^2 + \frac{1}{2\alpha_k} r_k^2 . \end{aligned} \tag{8.60}$$

Integrating both sides of eq.(8.60) yields

$$V(t) - V(0) \leq -\frac{\alpha_k}{2}\int_0^t \epsilon_k^2(\tau)d\tau + \frac{1}{2\alpha_k}\int_0^t r_k^2(\tau)d\tau \ . \tag{8.61}$$

Let

$$a \triangleq \frac{2}{\alpha_k}\sup_{t>0}\{V(0) - V(t)\} \ . \quad b \triangleq \frac{1}{\alpha_k^2} \ . \tag{8.62}$$

then

$$\begin{aligned}\int_0^t \epsilon_k^2(\tau)d\tau &\leq \frac{2}{\alpha_k}[\ V(0) - V(t)\ ] + \frac{1}{\alpha_k^2}\int_0^t r_k^2(\tau)d\tau \\ &\leq a + b\int_0^t r_k^2(\tau)d\tau \ . \end{aligned} \tag{8.63}$$

Since $(\int_0^\infty \| r_k(\tau) \|^2 d\tau)^{\frac{1}{2}} < \infty$, we have

$$(\int_0^\infty \epsilon_k^2(\tau)d\tau)^{\frac{1}{2}} \leq (a + b\int_0^\infty r_k^2(\tau)d\tau)^{\frac{1}{2}} < \infty \ . \tag{8.64}$$

i.e., $\epsilon_k(t) \in \mathcal{L}_2$. By result 8.2.2, we know that $\epsilon_k(t)$ is bounded, i.e., $\epsilon_k(t) \in \mathcal{L}_\infty$. We thus have $\epsilon_k(t) \in (\mathcal{L}_2 \bigcap \mathcal{L}_\infty)$. Since r_k is bounded, the boundedness of $\dot{\epsilon}_k$ can be concluded from eq.(8.32), i.e., $\dot{\epsilon}_k(t) \in \mathcal{L}_\infty$. By Corollary 2.9 in [114], we conclude that

$$\lim_{t\to\infty} |\ \epsilon_k(t)\ | = 0 \ . \tag{8.65}$$

This completes the proof.

8.3.4 Remarks

1. If the output of the identifier is used as one of the system inputs to replace the input variable it estimates, i.e.,

$$\dot{y}_k = -\alpha_k y_k + \mathbf{\Theta}_k^T(\mathbf{w}, \mathbf{u})\overline{\mathbf{Y}}_k \ , \ \alpha_k > 0, \ \ k \in \{1, \ldots, N\}. \tag{8.66}$$

where $\mathbf{w} \triangleq \{x_1, \ldots, x_{k-1}, y_k, x_{k+1}, \ldots, x_N\}^T$, we term the system a *recurrent dynamic fuzzy logic system*, which will be elaborated on later in this work. To

distinguish this from the DFLS of eq.(8.23), we term the latter a *non-recurrent DFLS*. For the sake of simplicity we will generally omit the label *non-recurrent*.

2. The development of a DFLS identifier consists of off-line design (for Θ_k) and on-line training (for $\overline{\mathbf{Y}}_k$). In off-line design, the number of primary fuzzy sets for each universe of discourse, as well as the positions and shapes of membership functions for each primary fuzzy set are to be determined. This is a very active but still immature area of research. Thus far, *ad hoc* analysis based on physical intuition is still required. After off-line design, the parameter vector $\overline{\mathbf{Y}}_k$ is trained on-line with the training law given in eq.(8.34). If the results are unsatisfactory, one must return to the off-line design process and repeat the whole procedure.

3. The importance of the initial values in nonlinear problems is well known, and an important point yet remains to be addressed, viz.,
   What is an appropriate initial value for vector $\overline{\mathbf{Y}}_k$ before training begins?
   This brings out a very important feature of a fuzzy logic system. A human being can generally acquire some knowledge of the behavior of almost any physical process through experience, which may be described linguistically in the form of IF-THEN rules. A fuzzy logic system provides a theoretical framework into which fuzzy linguistic information can be incorporated and processed systematically, a process for which powerful mathematical tools are available. The initial values of parameters $\overline{y}_i$, $i = 1, \ldots, I$, can therefore be determined from the initial IF-THEN rules, and further revised or refined through on-line training processes.

4. For some processes, if there is *enough* human expertise available, a fuzzy logic system may perform well enough that no further training is necessary, e.g., the fuzzy logic systems presented in Chapter 6 for simple control tasks. If

there is no human expertise available at all, the initial values must be assigned randomly. Nevertheless, the DFLS identifier's performance is as good as specified in Theorem 8.2. If there is only limited human expertise available, which is the usual situation in practice, the initial values based on this limited expertise are generally better than those randomly assigned and, in turn, the system may require less training and display significantly improved behavior. This will be illustrated through simulation experiments in the next section.

5. It is interesting to compare the DFLS with the *High Order Neural Network* presented by Kosmatopoulos et al. [81]. They share similar mathematical expressions in the input-output sense, and they both possess learning abilities. However, they originated from different physical backgrounds, and they have different system structures and physical perspectives. As pointed out by Wang [184], artificial neural networks try to emulate the *hardware* of the human brain, whereas fuzzy systems try to emulate the *software* in the human brain. The ANN is characterized with massive parallelism, but its parameters lack explicit physical meaning. The fuzzy logic system provides an effective framework to utilize human linguistic descriptions of unknown systems, and its parameters have clear physical meaning. In addition, the similarities in their mathematical expressions indicate that if there were no initial human expertise integrated into a fuzzy logic system, the DFLS would perform about as well as a high order neural network.

6. Comparing the DFLS with the static FLS, e.g., those presented in [88, 177], the dynamic feature of DFLS creates the potential to take advantage of the intrinsic dynamics of physical processes, and to thus be integrated into dynamic systems more naturally. The DFLS thus provides us with new tools in dealing with dynamic systems.

8.4 Identification of Nonlinear Systems – Illustrative Applications of DFLS

8.4.1 Example 1 – A Nonlinear System with Chaotic Behavior

Consider a nonlinear system

$$\ddot{x} + 0.1\dot{x} + x^5 = 6\sin t \ . \tag{8.67}$$

which exhibits chaotic behavior and may represent a lightly damped, sinusoidally forced mechanical structure undergoing large elastic deflections [151]. Let

$$\begin{cases} x_1 \ \hat{=} \ x \ , \\ x_2 \ \hat{=} \ \dot{x} \ , \end{cases} \tag{8.68}$$

and

$$\mathbf{x}(t) \ \hat{=} \ \begin{Bmatrix} x_1(t) \\ x_2(t) \end{Bmatrix} \ . \tag{8.69}$$

Then eq.(8.67) is expressed in state space form

$$\begin{cases} \dot{x}_1 \ = \ x_2 \ , \\ \dot{x}_2 \ = \ -x_1^5 - 0.1x_2 + 6\sin t \ . \end{cases} \tag{8.70}$$

The phase plane trajectory of this system with initial condition $\mathbf{x}(0) \ \hat{=} \ \{0, \ 0\}^T$ is shown in fig.8.2 for $t = 0$ to $t = 70$ sec.

Our objective here is to identify the state variables, x_1 and x_2, using the DFLS based identification algorithm developed in previous sections. It should be stressed that the dynamic model will be treated strictly as a black box, i.e., it is unknown to the identifiers.

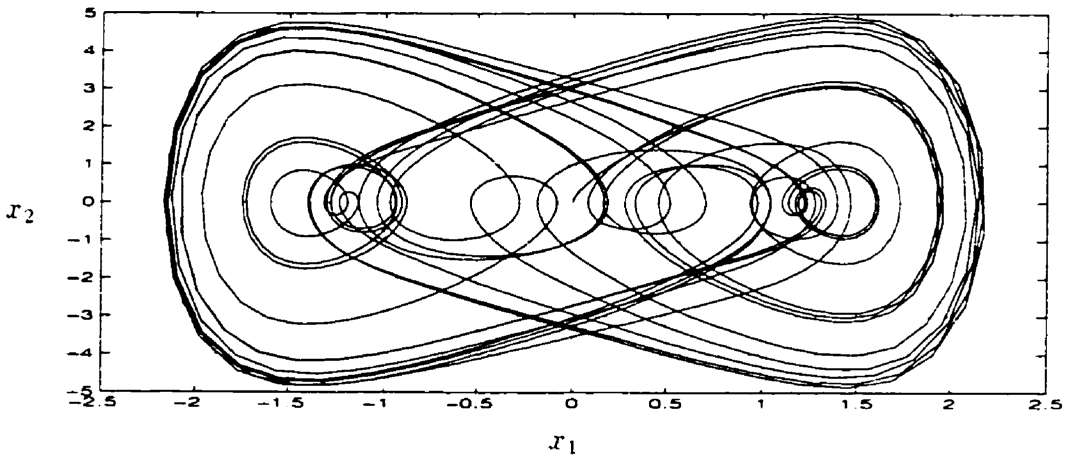


Figure 8.2: Phase Plane Trajectory - Example 1

Two DFLS identifiers, namely, D_{x_1} and D_{x_2}, are used for x_1 and x_2, respectively, i.e.,

$$\begin{cases} y_1 = D_{x_1}(\mathbf{x}) , \\ y_2 = D_{x_2}(\mathbf{x}) , \end{cases} \tag{8.71}$$

where y_1 and y_2 are the estimates of x_1 and x_2, respectively, with estimation errors, e_1 and e_2, defined as

$$\begin{cases} e_1 \triangleq y_1 - x_1 , \\ e_2 \triangleq y_2 - x_2 , \end{cases} \tag{8.72}$$

A DFLS Identifier for x_1 : D_{x_1}

D_{x_1} has two inputs, $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$, and one output, $y_1 \in \mathcal{Y}_1$, where $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{Y}_1$ are universes of discourse of linguistic variables, x_1, x_2 and y_1, respectively. In both $\mathcal{X}_1$ and $\mathcal{X}_2$, five primary fuzzy sets, A_{1j_1} and A_{2j_2}, $j_1 = 1,\ldots,5$, $j_2 = 1,\ldots,5$, are defined. For greater linguistic appeal, let

$$\begin{cases} \{A_{11}, A_{12}, A_{13}, A_{14}, A_{15}\}^T \triangleq \{(NZ)_1, (NS)_1, (ZR)_1, (PS)_1, (PZ)_1\}^T, \\ \\ \{A_{21}, A_{22}, A_{23}, A_{24}, A_{25}\}^T \triangleq \{(NZ)_2, (NS)_2, (ZR)_2, (PS)_2, (PZ)_2\}^T, \end{cases} \tag{8.73}$$

where P means positive, N means negative, and ZR stands for zero. Gaussian type membership functions are used for the primary fuzzy sets. For $p = 1, 2$,

$$\begin{aligned}
\mu_{A_{p1}}(x'_p) &\triangleq \begin{cases} 1 . & if \quad x'_p < \overline{x}_{p1} . \\ \exp[-\frac{1}{2}(\frac{x'_p - \overline{x}_{p1}}{\sigma_{p1}})^2] . & if \quad x'_p \geq \overline{x}_{p1} . \end{cases} \\
\mu_{A_{pj_p}}(x'_p) &\triangleq \exp[-\frac{1}{2}(\frac{x'_p - \overline{x}_{pj_p}}{\sigma_{pj_p}})^2] . \quad j_1, j_2 = 2, 3, 4. \\
\mu_{A_{p5}}(x'_p) &\triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{x'_p - \overline{x}_{p5}}{\sigma_{p5}})^2] . & if \quad x'_p < \overline{x}_{p5} . \\ 1 . & if \quad x'_p \geq \overline{x}_{p5} . \end{cases}
\end{aligned} \tag{8.74}$$

where the shape parameters of all the primary fuzzy sets are defined to be 0.45, i.e.,

$$\sigma_{pj_p} \triangleq 0.45 . \tag{8.75}$$

and the position parameters, $\overline{x}_{pj_p}$, are, for $p = 1, 2$,

$$\{\overline{x}_{p1}, \overline{x}_{p2}, \overline{x}_{p3}, \overline{x}_{p4}, \overline{x}_{p5}\}^T \triangleq \{-2, -1, 0, 1, 2\}^T . \tag{8.76}$$

The primary fuzzy sets are illustrated in fig.8.3.

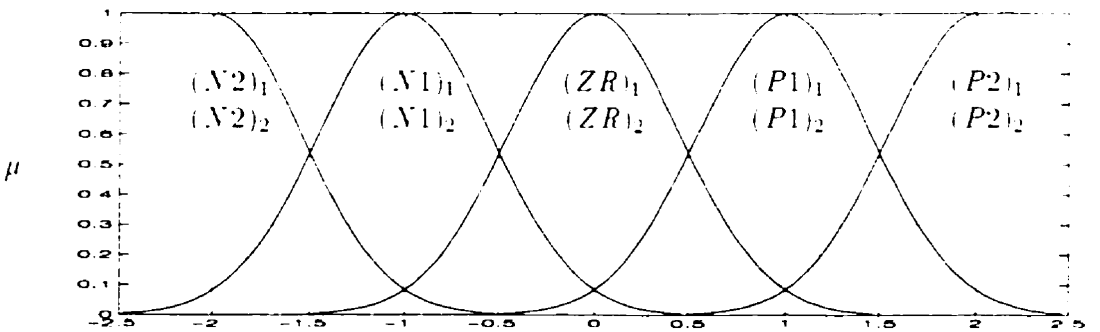


Figure 8.3: Primary Fuzzy Sets in $\mathcal{X}_1$ and $\mathcal{X}_2$ for D_{r_2}

In eq.(8.74), x'_p, $p = 1, 2$, are filtered values of inputs, x_p. The prefilters are defined as

$$\begin{cases} x'_1 = f_1(x_1) \triangleq x_1 . \\ x'_2 = f_2(x_2) \triangleq x_2 \cdot \frac{1}{2} . \end{cases} \tag{8.77}$$

which transform the majority of data of both variables, x_1 and x_2, into $[-2, 2]$.

The output of D_{x_1}, y_1, which is the estimate of x_1, is defined with the DFLS of eq.(8.5), i.e.,

$$\dot{y}_1 = -\alpha_1 y_1 + \mathbf{\Theta}_1^T \overline{\mathbf{Y}}_1 \ . \tag{8.78}$$

In this equation,

$$\begin{cases} \mathbf{\Theta}_1(\mathbf{x}') \triangleq \{\theta_1^1(\mathbf{x}'),\ \theta_2^1(\mathbf{x}'),\ \ldots.\ \theta_{25}^1(\mathbf{x}')\}^T \ . \\ \overline{\mathbf{Y}}_1 \triangleq \{\overline{y}_1^1,\ \overline{y}_2^1,\ \ldots.\ \overline{y}_{25}^1\}^T \ . \end{cases} \tag{8.79}$$

where $\mathbf{x}' \triangleq \{x_1', x_2'\}^T$, and

$$\theta_i^1(\mathbf{x}') = \frac{\prod_{p=1}^{2} \mu_{A_p^i}(x_p')}{\sum_{i=1}^{25} \prod_{p=1}^{2} \mu_{A_p^i}(x_p')} \ . \tag{8.80}$$

and for $i = 1, \ldots, 25$,

$$\begin{cases} A_1^i \in \{A_{1j_1}:\ j_1 = 1, \ldots, 5\} \ . \\ A_2^i \in \{A_{2j_2}:\ j_2 = 1, \ldots, 5\} \ . \end{cases} \tag{8.81}$$

The positive constant, α_1, is set to be 10. Now, $\overline{\mathbf{Y}}_1$ is left as a free parameter vector that is to be adaptively adjusted with the training law, eq.(8.34), i.e.,

$$\dot{\overline{\mathbf{Y}}}_1 = -\mathbf{H}_1 \mathbf{\Theta}_1 \epsilon_1 - S_1 \beta_1 \mathbf{H}_1 \overline{\mathbf{Y}}_1 \ . \tag{8.82}$$

In computer implementation, the derivative is approximated by a difference, i.e.,

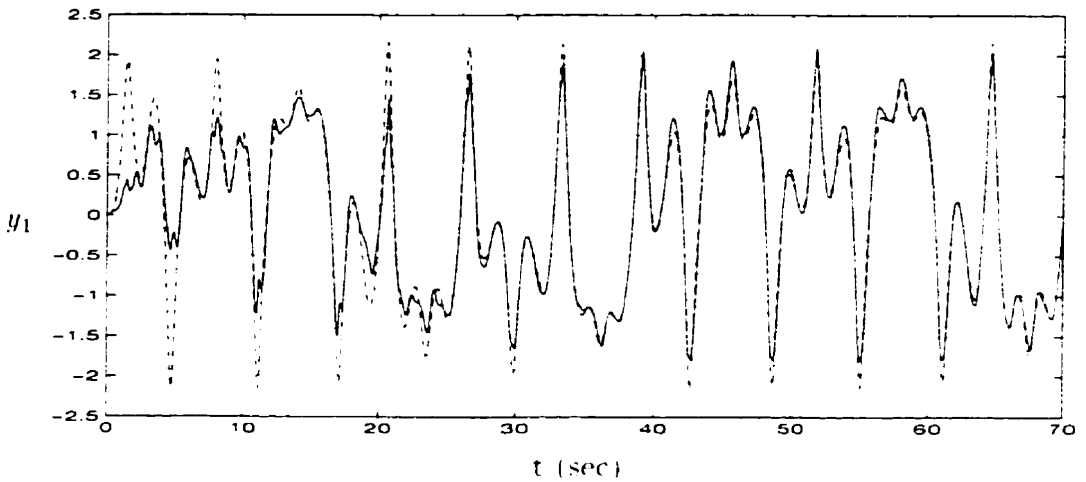
$$\dot{\overline{\mathbf{Y}}}_1 \triangleq \frac{\overline{\mathbf{Y}}_1(kT + T) - \overline{\mathbf{Y}}_1(kT)}{T} \ , \qquad k = 1, 2, \ldots \tag{8.83}$$

where $T \triangleq 0.05$ is the time incremental step. Use of eq.(8.83) in eq.(8.82) yields

$$\overline{\mathbf{Y}}_1(kT + T) = \overline{\mathbf{Y}}_1(kT) - T[\mathbf{H}_1 \mathbf{\Theta}_1 \epsilon_1 + S_1 \beta_1 \mathbf{H}_1 \overline{\mathbf{Y}}_1(kT)] \ . \tag{8.84}$$

The initial values of the elements of the parameter vector, $\overline{\mathbf{Y}}_1(0)$, are assigned random numbers uniformly distributed in $(-1, 1)$. The gain matrix, $\mathbf{H}_1$, is defined as

a diagonal matrix with all diagonal elements equal to 60. this being determined after considerable trial and error as providing good results. The bound of the parameter vector, $M_{\overline{\mathbf{Y}}_1}$, is set to 10^3. Assume the system to be identified is initially at rest, i.e., $\mathbf{x}(0) = \{0.\ 0\}^T$. The identification processes for x_1 is shown in fig.8.4, where training ends at t=40 sec. The solid line represents identifier output, and the dashed line represents the real system output. It is seen that the training process converges very well, and after training ends, the identifier predicts the system state, x_1, quite well.



(—— Identifier Output; - - - - Plant Output)

(Training Terminates at t = 40 sec)

Figure 8.4: Identification of x_1 – Example 1

B DFLS Identifier for x_2 : D_{x_2}

D_{x_2} is designed similarly to D_{x_1}. It has also two inputs, $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$ and one output, $y_2 \in \mathcal{Y}_2$. Here, for both $\mathcal{X}_1$ and $\mathcal{X}_2$, nine primary fuzzy sets, B_{1_j}, and

B_{2j_2}, $j_1 = 1,\ldots,9$, $j_2 = 1,\ldots,9$, are defined. For more linguistic appeal, let

$$\begin{cases} \{B_{11}, B_{12}, B_{13}, B_{14}, B_{15}, B_{16}, B_{17}, B_{18}, B_{19}\}^T \triangleq \\ \quad \{(N4)_1, (N3)_1, (N2)_1, (N1)_1, (ZR)_1, (P1)_1, (P2)_1, (P3)_1, (P4)_1\}^T, \\ \{B_{21}, B_{22}, B_{23}, B_{24}, B_{25}, B_{26}, B_{27}, B_{28}, B_{29}\}^T \triangleq \\ \quad \{(N4)_2, (N3)_2, (N2)_2, (N1)_2, (ZR)_2, (P1)_2, (P2)_2, (P3)_2, (P4)_2\}^T, \end{cases} \tag{8.85}$$

Gaussian type membership functions are used for the primary fuzzy sets, i.e., for $p = 1, 2$.

$$\begin{aligned} \mu_{B_{p1}}(x'_p) &\triangleq \begin{cases} 1, & if \quad x'_p < \bar{x}_{p1}, \\ \exp[-\frac{1}{2}(\frac{x'_p - \bar{x}_{p1}}{\sigma_{p1}})^2], & if \quad x'_p \geq \bar{x}_{p1}, \end{cases} \\ \mu_{B_{pj_p}}(x'_p) &\triangleq \exp[-\frac{1}{2}(\frac{x'_p - \bar{x}_{pj_p}}{\sigma_{pj_p}})^2], \quad j_1, j_2 = 2,\ldots,8, \\ \mu_{B_{p9}}(x'_p) &\triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{x'_p - \bar{x}_{p9}}{\sigma_{p9}})^2], & if \quad x'_p < \bar{x}_{p9}, \\ 1, & if \quad x'_p \geq \bar{x}_{p9}, \end{cases} \end{aligned} \tag{8.86}$$

where the shape parameters of all the primary fuzzy sets are defined to be 0.45, i.e.,

$$\sigma_{pj_p} \triangleq 0.45, \tag{8.87}$$

and the position parameters, $\bar{x}_{pj_p}$, are defined as, for $p = 1, 2$,

$$\{\bar{x}_{p1}, \bar{x}_{p2}, \bar{x}_{p3}, \bar{x}_{p4}, \bar{x}_{p5}, \bar{x}_{p6}, \bar{x}_{p7}, \bar{x}_{p8}, \bar{x}_{p9}\}^T \triangleq \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}^T. \tag{8.88}$$

The primary fuzzy sets are illustrated in fig.8.5.

In eq.(8.86), x'_p, $p = 1, 2$, are filtered values of inputs, x_p. The prefilters are defined as

$$\begin{cases} x'_1 = f_1(x_1) \triangleq x_1 \cdot 2, \\ x'_2 = f_2(x_2) \triangleq x_2. \end{cases} \tag{8.89}$$

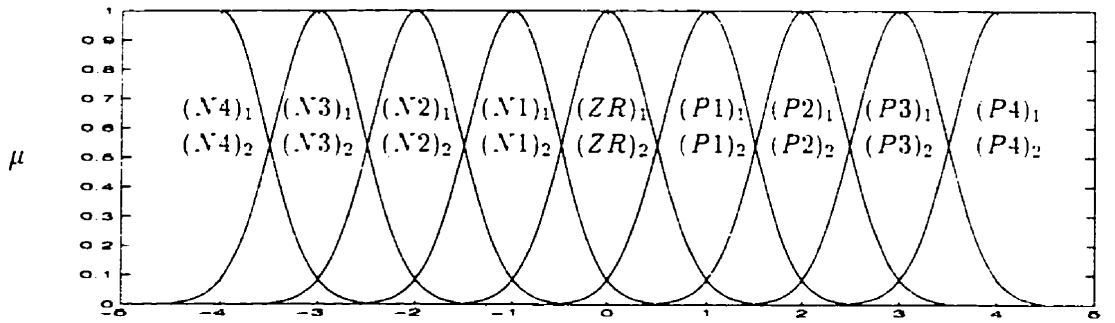


Figure 8.5: Primary Fuzzy Sets in $\mathcal{X}_1$ and $\mathcal{X}_2$ for D_{x_2}

which transform the majority of data of both variables, x_1 and x_2 into $[-4, 4]$.

The output of D_{x_2}, y_2, which is the estimate of x_2, is defined with the DFLS of eq.(8.5), i.e.,

$$\dot{y}_2 = -a_2 y_2 + \mathbf{\Theta}_2^T \overline{\mathbf{Y}}_2 \ . \tag{8.90}$$

In this equation,

$$\begin{cases} \mathbf{\Theta}_2(\mathbf{x}') & \triangleq & \{\theta_1^2(\mathbf{x}'),\ \theta_2^2(\mathbf{x}'),\ \ldots,\ \theta_{81}^2(\mathbf{x}')\}^T \ . \\ \overline{\mathbf{Y}}_2 & \triangleq & \{\overline{y}_1^2,\ \overline{y}_2^2,\ \ldots,\ \overline{y}_{81}^2\}^T \ . \end{cases} \tag{8.91}$$

where

$$\theta_i^2(\mathbf{z}) = \frac{\prod_{p=1}^{2} \mu_{B_p^i}}{\sum_{i=1}^{81} \prod_{p=1}^{2} \mu_{B_p^i}} \ , \qquad i = 1, \ldots, 81, \tag{8.92}$$

and for $i = 1, \ldots, 81$,

$$\begin{cases} B_1^i & \in & \{B_{1j_1}:\ j_1 = 1, \ldots, 9\} \ . \\ B_2^i & \in & \{B_{2j_2}:\ j_2 = 1, \ldots, 9\} \ . \end{cases} \tag{8.93}$$

The free parameter vector, $\overline{\mathbf{Y}}_2$, is adaptively adjusted with the training law, eq.(8.34), i.e.,

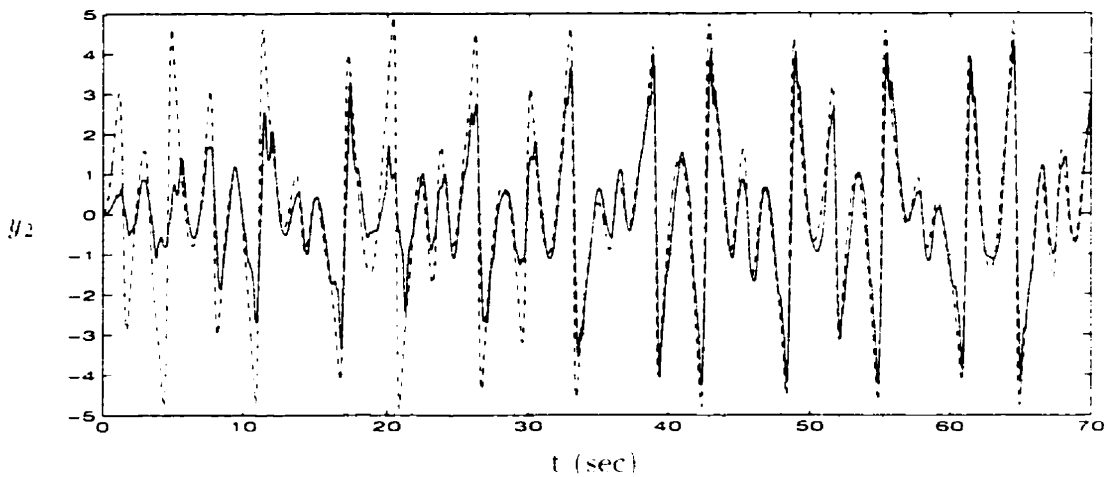
$$\dot{\overline{\mathbf{Y}}}_2 = -\mathbf{H}_2 \mathbf{\Theta}_2 e_2 - S_2 \beta_2 \mathbf{H}_2 \overline{\mathbf{Y}}_2 \ . \tag{8.94}$$

Approximating the derivative by a difference yields

$$\overline{\mathbf{Y}}_2(kT + T) = \overline{\mathbf{Y}}_2(kT) - T[\mathbf{H}_2 \mathbf{\Theta}_2 e_2 + S_2 \beta_2 \mathbf{H}_2 \overline{\mathbf{Y}}_2(kT)] \ . \tag{8.95}$$

where T is the same as in the case of D_{x_1}.

The initial values of the elements of the parameter vector, $\overline{\mathbf{Y}}_2(0)$, are also assigned random numbers uniformly distributed in (-1, 1). We take the positive constant $\alpha_2 \triangleq 10$. The gain matrix, $\mathbf{H}_2$, is defined as a diagonal matrix with all diagonal elements equal to 60. The bound of the parameter vector, $M_{\overline{\mathbf{Y}}_2}$, is set to 10^3. With the system to be identified initially at rest, the identification processes for x_2 is shown in fig.8.6, where training ends at t=40 sec. The solid line represents identifier output, and dashed line represents real system output. Again, the training process converges very well, and after training ends, the identifier predicts system state, x_2, quite well.



(—— Identifier Output; ---- Plant Output)

(Training Terminates at t = 40 sec)

Figure 8.6: Identification of x_2 - Example 1

C Test of Trained DFLS Identifiers

Both identifiers, D_{x_1} and D_{x_2}, were able to predict the system states quite well, demonstrating that the system was quite well identified for the specific initial condition.

For chaotic systems, the choice of initial conditions has a significant effect on the system dynamics, as illustrated in fig.8.7, where the solid lines represent the dynamics for $\mathbf{x}(0) = \{2,\ 3\}^T$, and the dashed lines represent the dynamics for $\mathbf{x}(0) = \{0,\ 0\}^T$.

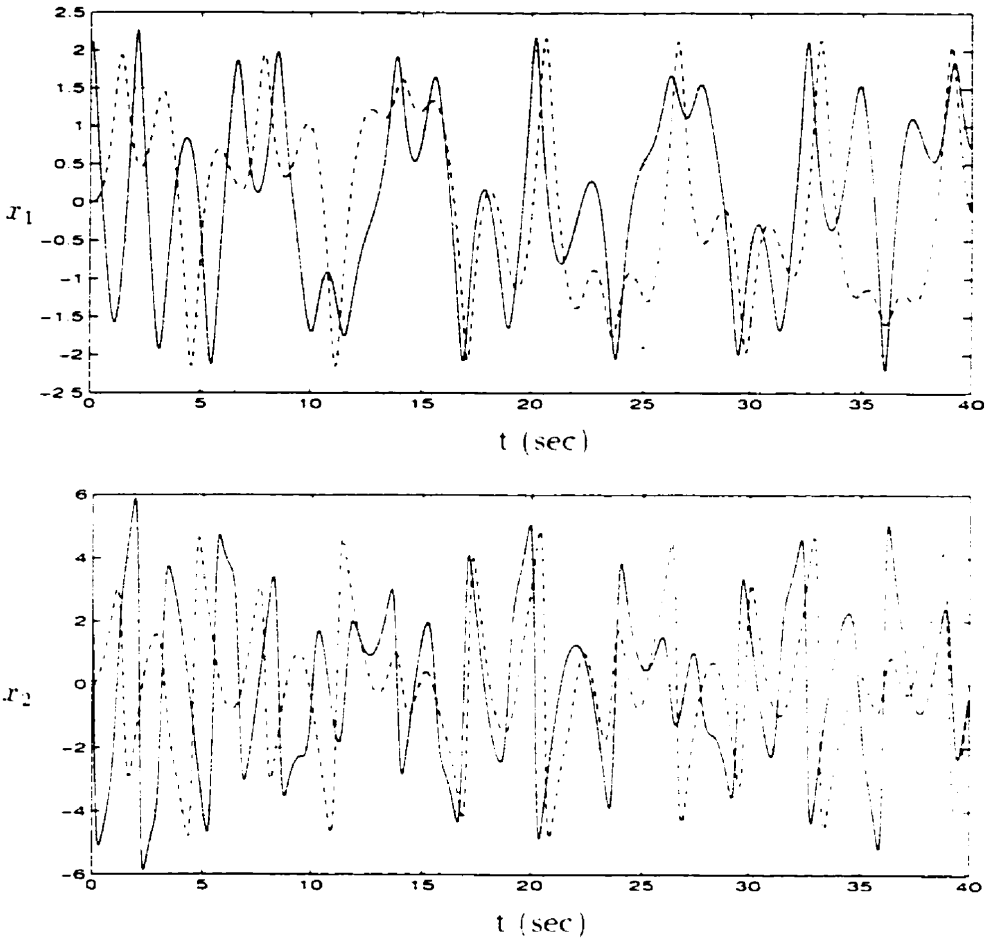


Figure 8.7: System Dynamics for Different Initial Conditions

Next the trained identifiers, D_{x_1} and D_{x_2}, are tested against a different initial condition, $\mathbf{x}(0) = \{2,\ 3\}^T$. The identification results are shown in fig.8.8, where the solid lines represent identifier outputs, and the dashed lines represent real system outputs. Again, the results are quite satisfactory and indicate that the DFLS based

identification algorithm is very effective in dealing with nonlinear systems. It should

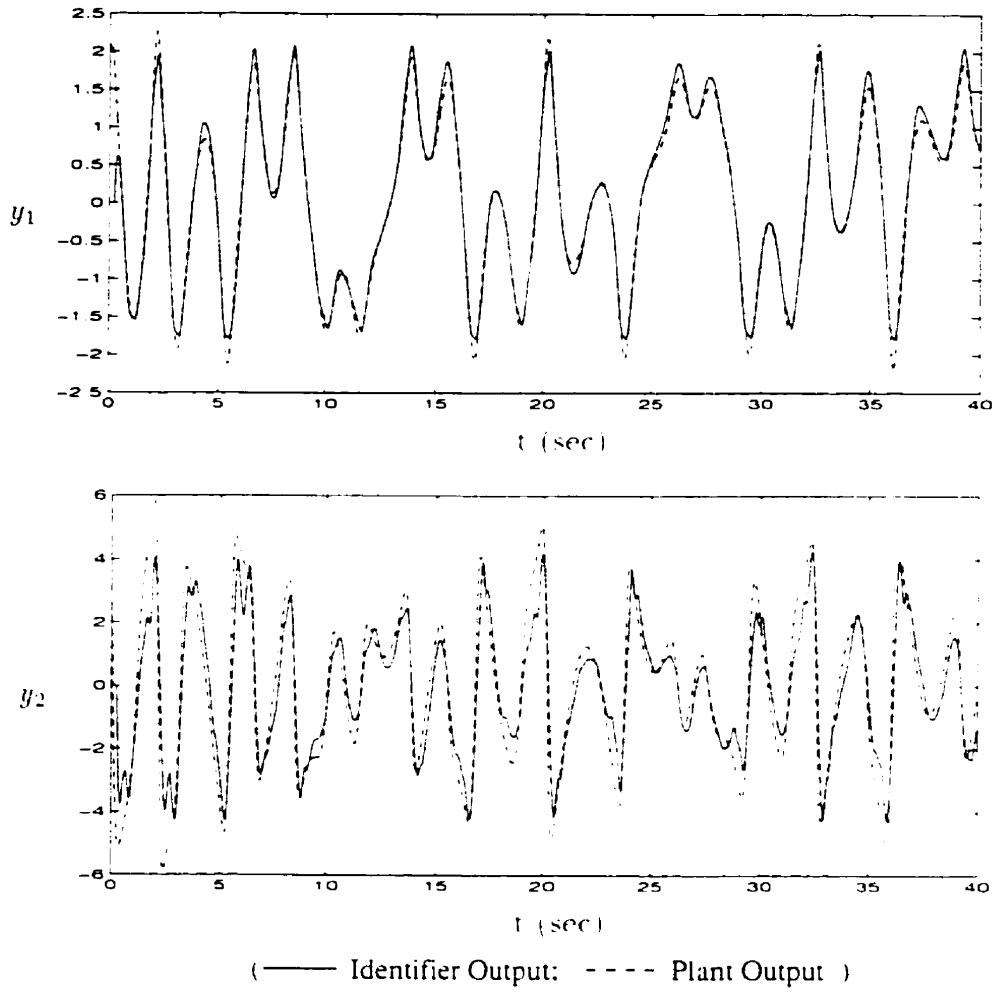


Figure 8.8: Test of Trained Identifiers for $\mathbf{x}(0) = \{2, 3\}^T$

be mentioned that all simulations in this work are programmed with MATLAB and SIMULINK.

8.4.2 Example 2 – Chaotic Glycolytic Oscillator

We now demonstrate another identification example, and compare our results with that of a static FLS.

Consider the following chaotic glycolytic oscillator [184, chap.11].

$$\begin{cases} \dot{x}_1(t) &= -x_1(t)x_2^2(t) + 0.999 + 0.42\cos(1.75t) \\ \dot{x}_2(t) &= x_1(t)x_2^2(t) - x_2(t) \end{cases} \tag{8.96}$$

Our objective is to identify the state variables, x_1 and x_2, using the DFLS based identification algorithm. As in the previous simulation examples, the system dynamic model is strictly treated as a black box.

Two DFLS identifiers, G_{x_1} and G_{x_2}, are used to identify the two state variables, x_1 and x_2, respectively, i.e.,

$$\begin{cases} y_1 &= G_{x_1}(\mathbf{x}) \ , \\ y_2 &= G_{x_2}(\mathbf{x}) \ , \end{cases} \tag{8.97}$$

where y_1 and y_1 represent estimates of the state variables, x_1 and x_2, respectively. The identification errors, e_1 and e_2, are defined as

$$\begin{cases} e_1 &\hat{=} \ y_1 - x_1 \ , \\ e_2 &\hat{=} \ y_2 - x_2 \ , \end{cases} \tag{8.98}$$

Both identifiers, G_{x_1} and G_{x_2}, have two inputs, $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$, and one output, $y_1 \in \mathcal{Y}_1$ for G_{x_1} and $y_2 \in \mathcal{Y}_2$ for G_{x_2}.

A DFLS Identifier for x_1 : G_{x_1}

In both $\mathcal{X}_1$ and $\mathcal{X}_2$, five primary fuzzy sets, A_{1j_1} and A_{2j_2}, $j_1 = 1, \ldots, 5$, $j_2 = 1, \ldots, 5$, are defined. Their membership functions are those defined in eqs.(8.74-8.76), and illustrated in fig.8.3.

The phase plane trajectory of the system with initial condition, $\mathbf{x}(0) \hat{=} \{1.5, \ 1.5\}^T$, is shown in fig.8.9 for $t = 0$ to $t = 100$ sec. from which it is observed that most of the values of x_1 and x_2 are in [0.2, 3] and [0.3, 3], respectively. To transform these to [-2, 2], the following prefilters for x_1 and x_2 are defined

$$\begin{cases} x_1' &= f_1(x_1) \ \hat{=} \ (x_1 - \frac{3+0.2}{2}) \cdot \frac{2}{0.5 \cdot (3-0.2)} \ , \\ x_2' &= f_2(x_2) \ \hat{=} \ (x_2 - \frac{3+0.3}{2}) \cdot \frac{2}{0.5 \cdot (3-0.3)} \ , \end{cases} \tag{8.99}$$

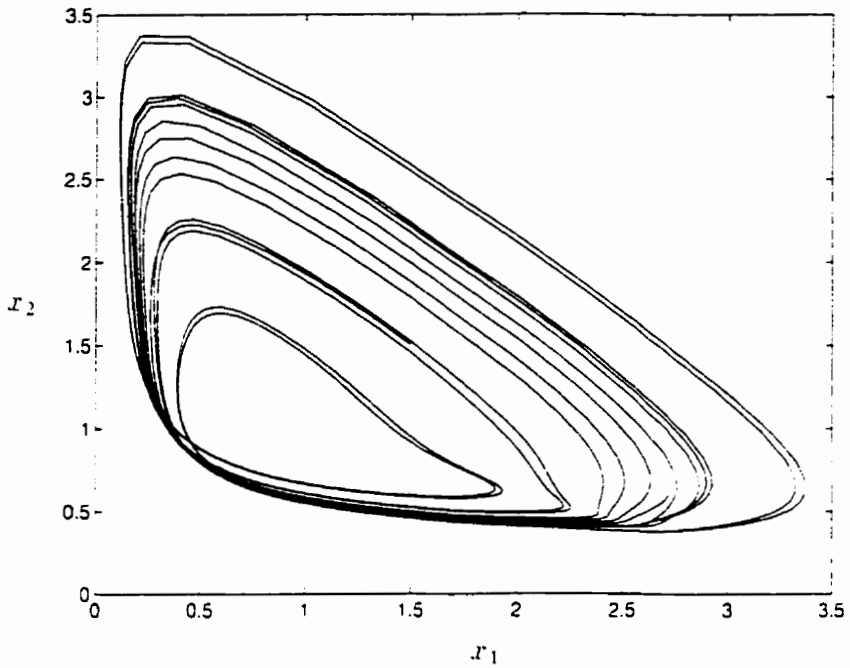


Figure 8.9: Phase Plane Trajectory - Example 2

The output of G_{x_1}, y_1, which is the estimate of x_1, is defined via the DFLS of eq.(8.5), i.e.,

$$\dot{y}_1 = -\alpha_1 y_1 + \boldsymbol{\Theta}_1^T \overline{\mathbf{Y}}_1 \ . \tag{8.100}$$

where $\boldsymbol{\Theta}_1$ and $\overline{\mathbf{Y}}_1$ are the same as those defined in eqs.(8.79-8.81).

The parameter vector, $\overline{\mathbf{Y}}_1$, has 25 free parameters and is adaptively adjusted with the training law, eq.(8.84), i.e.,

$$\overline{\mathbf{Y}}_1(kT+T) = \overline{\mathbf{Y}}_1(kT) - T[\mathbf{H}_1 \boldsymbol{\Theta}_1 e_1 + S_1 \beta_1 \mathbf{H}_1 \overline{\mathbf{Y}}_1(kT)] \ . \tag{8.101}$$

The time increment, T, is set to 0.1 sec and constant, α_1, is set to 10. The initial values of the elements of the parameter vector, $\overline{\mathbf{Y}}_1(0)$, are assigned random values uniformly distributed in (-1, 1). The gain matrix, $\mathbf{H}_1$, is defined as a diagonal matrix with all diagonal elements equal to 50. The bound of the parameter vector, $M_{\overline{\mathbf{Y}}_1}$, is set to 10^3.

Assume the system to be identified is initially at $\mathbf{x}(0) = \{1.5,\ 1.5\}^T$. The identification processes for x_1 is shown in fig.8.10, where training ends at $t = 50$ sec. The solid line represents identifier output, and dashed line represents real system output. It is seen that the convergence speed and accuracy of the training process are quite satisfactory. After training, the identifier predicts system state, x_1, very well.

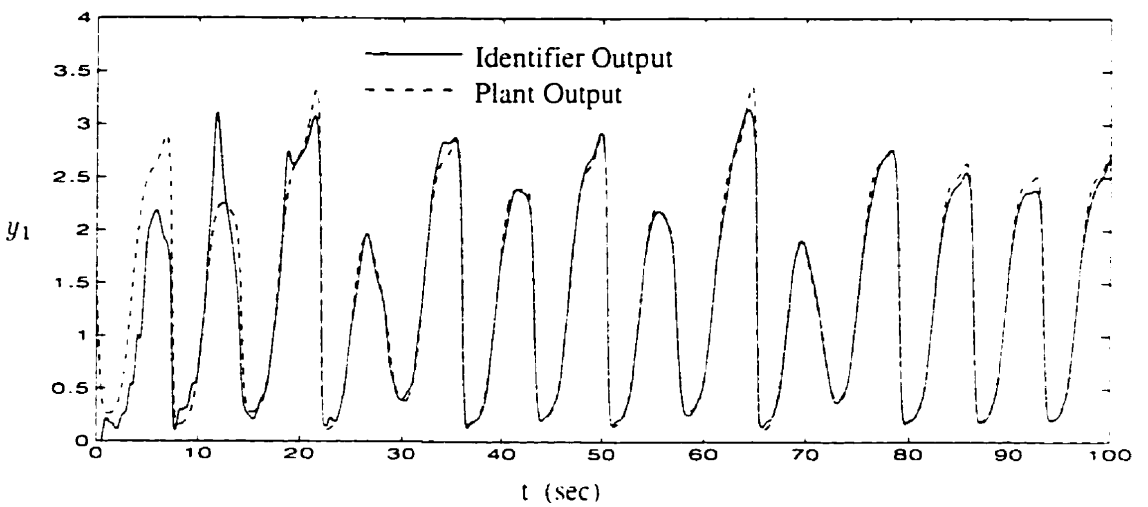


Figure 8.10: Identification of x_1 with $\mathbf{x}(0) = \{1.5,\ 1.5\}^T$ - Example 2

B DFLS Identifier for x_2 : G_{x_2}

The primary fuzzy sets in both $\mathcal{X}_1$ and $\mathcal{X}_2$ of G_{x_2} are defined to be the same as those of G_{x_1}, namely, A_{1j_1} and A_{2j_2}, $j_1 = 1,\ldots,5$, $j_2 = 1,\ldots,5$, and their membership functions those defined in eqs.(8.74–8.76).

The prefilters for x_1 and x_2 are the same as those defined in eq.(8.99), which transform the bulk of the values of both variables, x_1 and x_2, into $[-2,\ 2]$. The output of G_{x_2}, y_2, which is the estimate of x_2, is defined via the DFLS of eq.(8.5),

i.e..

$$\dot{y}_2 = -\alpha_2 y_2 + \Theta_2^T \overline{\mathbf{Y}}_2 \ . \tag{8.102}$$

where Θ_2 and $\overline{\mathbf{Y}}_2$ are defined in eqs.(8.79–8.81).

The parameter vector, $\overline{\mathbf{Y}}_2$, is adaptively adjusted with the training law, eq.(8.95, i.e..

$$\overline{\mathbf{Y}}_2(kT + T) = \overline{\mathbf{Y}}_2(kT) - T[\mathbf{H}_2\Theta_2\epsilon_2 + S_2\beta_2\mathbf{H}_2\overline{\mathbf{Y}}_2(kT)] \ . \tag{8.103}$$

The time increment, T, is set to 0.1 sec. and $\alpha_2 \triangleq 10$. The initial values of the elements of the parameter vector, $\overline{\mathbf{Y}}_2(0)$, are assigned random values uniformly distributed in (-1, 1). The gain matrix, $\mathbf{H}_2$, is defined as a diagonal matrix with all diagonal elements equal to 50, by trail and error as providing good results. The bound of the parameter vector, $M_{\overline{\mathbf{Y}}_2}$, is set to 10^3.

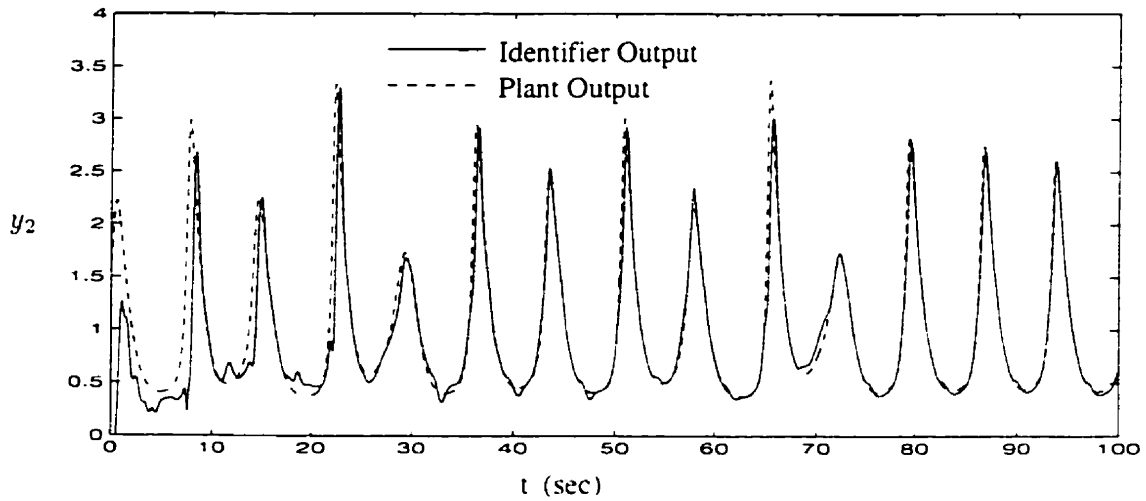
With the system to be identified initially at $\mathbf{x}(0) = \{1.5, 1.5\}^T$, the identification processes for x_2 is shown in fig.8.11, where training ends at $t = 50$ sec. The solid line represents identifier output, and dashed line represents real system output. Again, the convergence speed and accuracy of the training process are quite satisfactory, and after training stops, the identifier predicts system state, x_2, very well. These results, again, show the usefulness of this identification algorithm in dealing with complex nonlinear systems.

C Identification Results : DFLS vs. FLS

We now compare our DFLS based identification results with those obtained by means of the static FLS [184, chap.11].

The same plant, i.e. eq.(8.96), was identified in [184, chap.11] using static fuzzy logic systems. Figure 8.12 shows the identification results from this reference for x_1 and x_2 using so-called "first type" fuzzy identifiers[1]. Figure 8.13 shows their

[1]The kind of FLS which is linear in their adjustable parameters [184].



(Training Terminates at t=50 sec)

Figure 8.11: Identification of x_2 with $\mathbf{x}(0) = \{1.5,\ 1.5\}^T$ - Example 2

identification results for x_1 and x_2 using so-called "second type" fuzzy identifiers[2]. Figure 8.14 shows the identification results of x_1 and x_2 using our DFLS identifiers.

Comparing our DFLS results, fig.8.14, with those of FLS, figs.8.12–8.13, it is apparent that the DFLS identification results are superior to those of both first type FLS and second type FLS in terms of convergence accuracy.

[2]The kind of FLS which is nonlinear in their adjustable parameters [184].

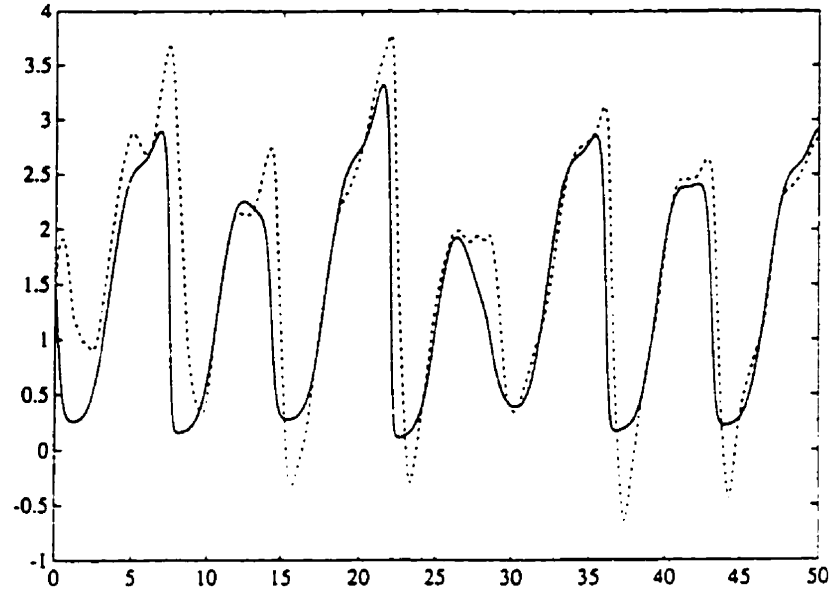


Figure 11.3 $x_1(t)$ (solid line) and $\hat{x}_1(t)$ (dashed line) using the first-type fuzzy identifier without incorporating any linguistic information.

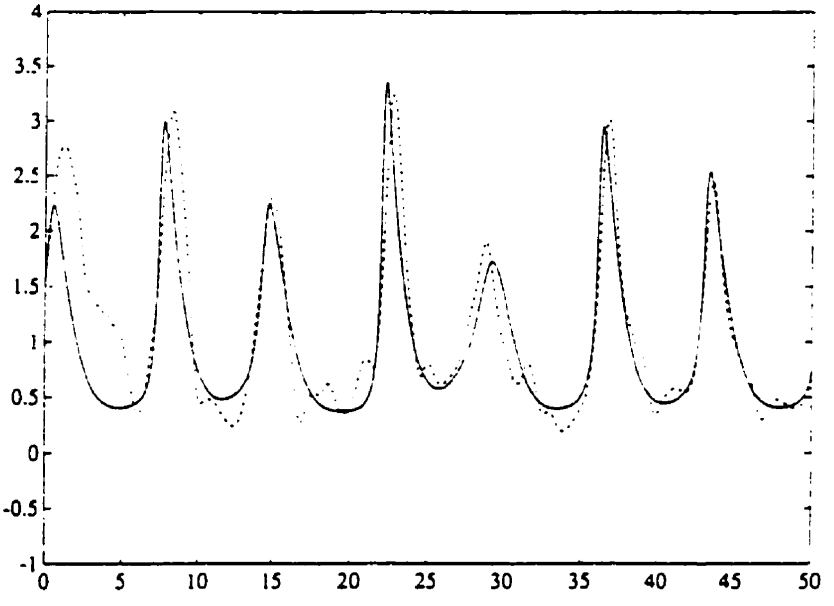


Figure 11.4 $x_2(t)$ (solid line) and $\hat{x}_2(t)$ (dashed line) using the first-type fuzzy identifier without incorporating any linguistic information.

Figure 8.12: Identification of x_1 and x_2 Using FLS ("First-Type Identifier")

(Identifier Output - Dashed Line: Plant Output - Solid Line)

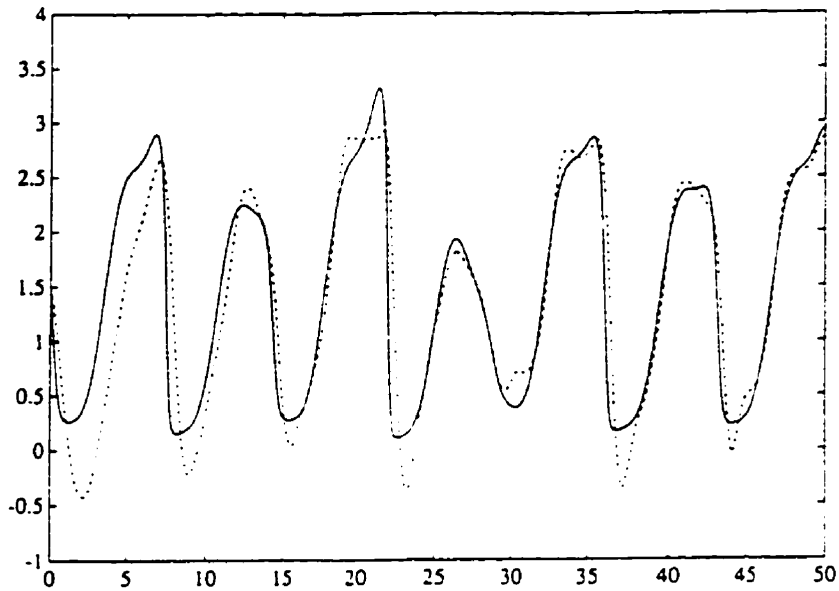


Figure 11.7 $x_1(t)$ (solid line) and $\hat{x}_1(t)$ (dashed line) using the second-type fuzzy identifier without incorporating any linguistic information.

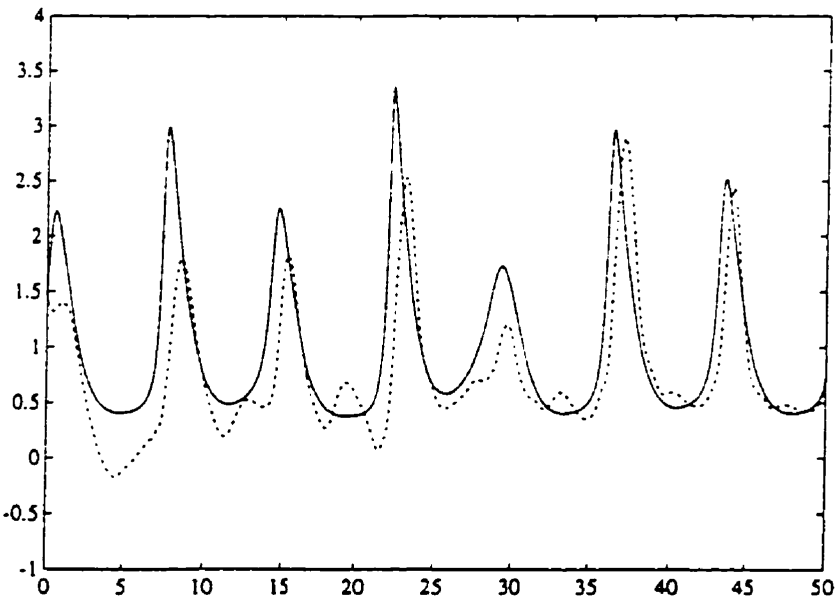
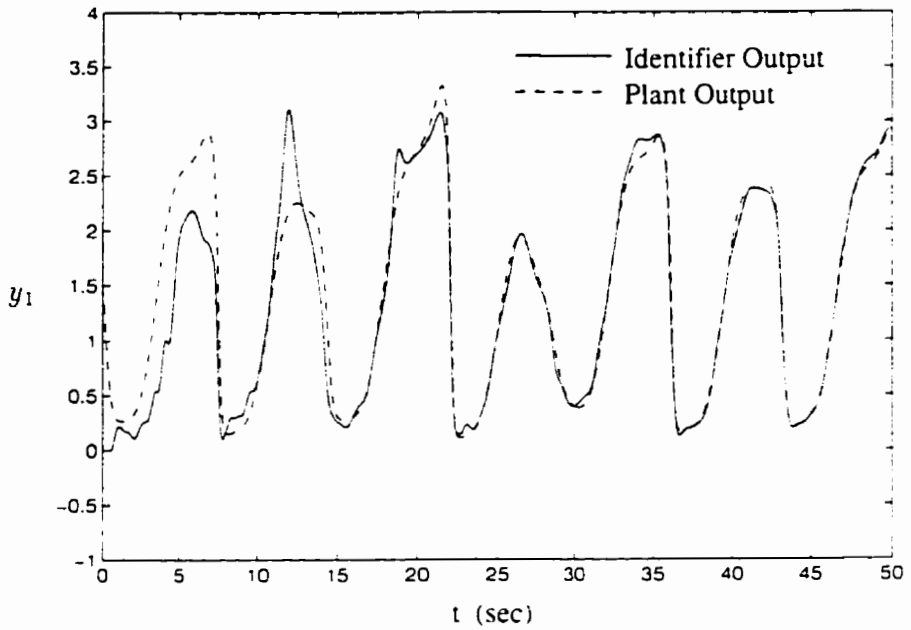
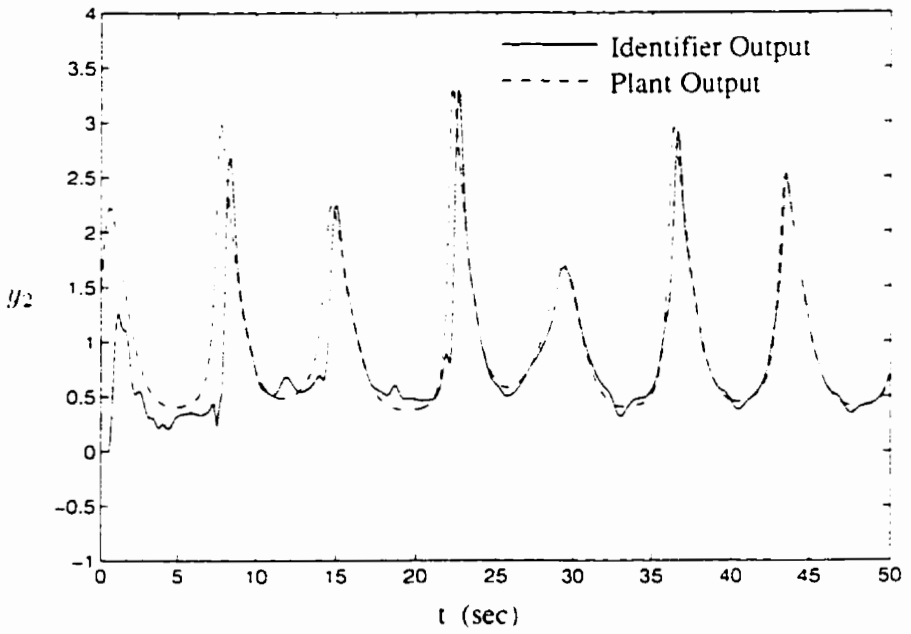


Figure 11.8 $x_2(t)$ (solid line) and $\hat{x}_2(t)$ (dashed line) using the second-type fuzzy identifier without incorporating any linguistic information.

Figure 8.13: Identification of x_1 and x_2 Using FLS ("Second-Type Identifier"; (Identifier Output - Dashed Line: Plant Output - Solid Line)



(a) Identification of x_1



(b) Identification of x_2

Figure 8.14: Identification of x_1 and x_2 Using DFLS (For Comparison with FLS Results)

8.4.3 Example 3 – A Flexible Link Manipulator with Non-linear Joint Friction

We now demonstrate the application of the DFLS identification algorithm to a practical system through simulation experiments.

Consider a flexible single link manipulator described by eq.(A.151) of Appendix A.

$$\begin{cases} \dot{X} &= AX + B(T_m - T_f) \,, \\ Y &= CX \,. \end{cases} \tag{8.104}$$

where T_m is the electro-magnetic torque of the motor, and T_f is the friction torque of the motor. The matrices, A, B, and C, are given in eqs.(A.141-A.143). The friction, T_f, is composed of stiction and dynamic friction, whose mathematical models are given in eqs.(A.145-A.150). Here, it is assumed that T_m is proportional to the command voltage applied to the motor, V_m, i.e.,

$$T_m = K_T V_m \,. \tag{8.105}$$

where K_T is the torque constant. The state vector, X, and output vector, Y, are defined in eq.(A.137) as

$$\begin{cases} X &\triangleq \{\, q_0,\ \dot{q}_0,\ q_1,\ \dot{q}_1,\ q_2,\ \dot{q}_2,\ q_3,\ \dot{q}_3,\ \}^T \,, \\ Y &\triangleq \{\, o(0,t),\ \dot{o}(0,t),\ v(L,t),\ \dot{v}(L,t),\ \}^T \,. \end{cases} \tag{8.106}$$

where q_i, $i = 0, \ldots, 3$, are generalized coordinates, $o(0,t)$ and $\dot{o}(0,t)$ are the angular displacement and the angular velocity at the root of the beam, $v(L,t)$ and $\dot{v}(L,t)$ are the deflection and its rate of change at the tip of the beam due to flexural vibration. Again, this dynamic model will be strictly treated as a black box from the point of view of the identifiers.

The beam is made of stainless steel. and has the following properties.

$$\begin{cases} \textit{length}: & L = 540 & mm. \\ \textit{width}: & H = 50 & mm. \\ \textit{thickness}: & B = 1.28 & mm. \\ \textit{mass density}: & \rho = 7.85 & g/cm^3. \\ \textit{Young's modulus}: & E = 206.7 & GN/m^2. \\ \textit{damping ratios}: & \xi_i = 0.002. & i = 1, 2, \ldots. \\ \textit{tip load}: & M_t = 0.246 & kg. \end{cases} \tag{8.107}$$

where mass density and Young's modulus are obtained from [23].

The actuator is a direct drive motor whose mass moment of inertia, J_h, and torque constant, K_T, are

$$\begin{cases} J_h &= 0.142 \quad kg \cdot m^2 \,, \\ K_T &= 18.75 \quad N \cdot m/V \,. \end{cases} \tag{8.108}$$

The parameters in the motor friction model are obtained from [3], and are as follows:

$$\begin{cases} \alpha_0 &= 3.5 & N \cdot m. \\ \alpha_1 &= 0.816 & N \cdot m. \\ \alpha_2 &= 0.876 & N \cdot m \cdot sec. \\ \tau_s &= 0.01 & sec. \\ \tau_r &= 0.001 & sec. \\ k_s &= 10000 & N \cdot m/rad. \\ d_s &= 1100 & N \cdot m \cdot sec/rad. \\ \dot{\alpha}_0(0,t) &= 0.04 & rad/sec. \end{cases} \tag{8.109}$$

Our objective was to identify the following variables using DFLS identifiers.

(i) angular displacement at the root of the beam, $\alpha(0,t)$;

(ii) angular velocity at the root of the beam, $\dot{\alpha}(0,t)$;

(iii) vibrational displacement at the tip of the beam, $v(L,t)$;

(iv) vibrational velocity at the tip of the beam, $\dot{v}(L,t)$.

Four DFLS identifiers were used, one for each variable to be identified, as illustrated in fig.8.15, where D_o for $o(0,t)$, $D_{\dot{o}}$ for $\dot{o}(0,t)$, D_v for $v(L,t)$, and $D_{\dot{v}}$ for $\dot{v}(L,t)$. Beam vibrational variables were not used in identifying $o(0,t)$ and $\dot{o}(0,t)$, because based on our experience, this increased the size of the fuzzy rule base and slowed down the fuzzy inference process which is crucial for on-line operation, and did little to improve accuracy. For the same reason, angular displacement of the motor hub, $o(0,t)$, was not used in identifying beam vibrational variables, $v(L,t)$ and $\dot{v}(L,t)$.

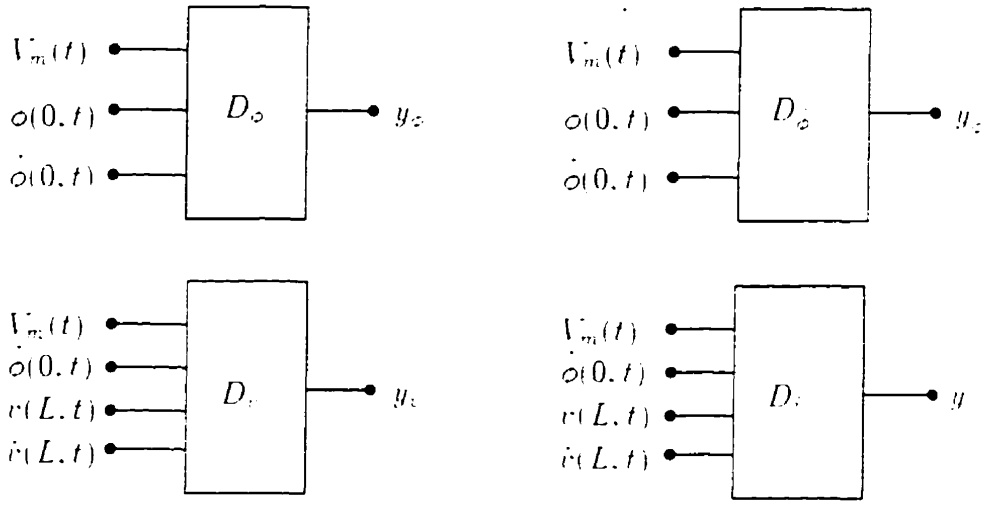


Figure 8.15: Inputs-Outputs of DFLS identifiers - Example 3

The identification errors are denoted as

$$\begin{cases} \epsilon_o \triangleq y_o - o(0,t) , \\ \epsilon_{\dot{o}} \triangleq y_{\dot{o}} - \dot{o}(0,t) , \\ \epsilon_v \triangleq y_v - v(L,t) , \\ \epsilon_{\dot{v}} \triangleq y_{\dot{v}} - \dot{v}(L,t) . \end{cases} \tag{8.110}$$

All the universes of discourse were partitioned into five primary fuzzy sets:

$$\{N2, N1, ZR, P1, P2\}.$$

whose membership functions were the same as those defined in eqs.(8.74-8.76) and illustrated in fig.8.3. Each input variable in fig.8.15 passed through a prefilter to transform the majority of their values into [-2, 2].

A DFLS Identifiers for $o(0,t)$ and $\dot{o}(0,t)$

With the DFLS of eq.(8.5),

$$\dot{y}_o = -\alpha_o y_o + \mathbf{\Theta}_o^T(\mathbf{z}'_o)\overline{\mathbf{Y}}_o \,. \tag{8.111}$$

$$\dot{y}_{\dot{o}} = -\alpha_{\dot{o}} y_{\dot{o}} + \mathbf{\Theta}_{\dot{o}}^T(\mathbf{z}'_{\dot{o}})\overline{\mathbf{Y}}_{\dot{o}} \,. \tag{8.112}$$

where

$$\mathbf{z}'_o \triangleq \{o'(0,t),\ \dot{o}'(0,t),\ V'_m\}^T \,. \tag{8.113}$$

The prefilters are defined as

$$\begin{cases} o'(0,t) \triangleq f_o[o(0,t)] \triangleq [o(0,t) - \frac{9+0}{2}] \cdot \frac{2}{0.5*(9-0)} \,. \\ \dot{o}'(0,t) \triangleq f_{\dot{o}}[\dot{o}(0,t)] \triangleq \dot{o}(0,t) \cdot \frac{2}{0.5*(5-5)} \,. \\ V'_m \triangleq f_{V_m}(V_m) \triangleq V_m \cdot 2 \,. \end{cases} \tag{8.114}$$

Both $\mathbf{\Theta}_o$ and $\mathbf{\Theta}_{\dot{o}}$ have 125 free parameters, and are trained using adaptive law, eq.(8.34). Use of a difference to approximate the derivative in eq.(8.34) gives

$$\begin{cases} \overline{\mathbf{Y}}_o(kT+T) = \overline{\mathbf{Y}}_o(kT) - T[\mathbf{H}_o\mathbf{\Theta}_o\epsilon_o + S_o\beta_o\mathbf{H}_o\overline{\mathbf{Y}}_o(kT)] \,. \\ \overline{\mathbf{Y}}_{\dot{o}}(kT+T) = \overline{\mathbf{Y}}_{\dot{o}}(kT) - T[\mathbf{H}_{\dot{o}}\mathbf{\Theta}_{\dot{o}}\epsilon_{\dot{o}} + S_{\dot{o}}\beta_{\dot{o}}\mathbf{H}_{\dot{o}}\overline{\mathbf{Y}}_{\dot{o}}(kT)] \,. \end{cases} \tag{8.115}$$

where the time increment, T, is set to 0.05 sec, the gain matrices, $\mathbf{H}_o$ and $\mathbf{H}_{\dot{o}}$, are diagonal matrices with all diagonal elements equal to 80, and the bounds of the parameter vectors, M_o and $M_{\dot{o}}$, are set to 10^3. The constants, α_o and $\alpha_{\dot{o}}$, are set to 10.

B DFLS Identifiers for $v(L,t)$ and $\dot{v}(L,t)$

With the DFLS of eq.(8.5),

$$\dot{y}_v = -\alpha_v y_v + \mathbf{\Theta}_v^T(\mathbf{z}_v')\overline{\mathbf{Y}}_v \ . \tag{8.116}$$

$$\dot{y}_{\dot{v}} = -\alpha_{\dot{v}} y_{\dot{v}} + \mathbf{\Theta}_{\dot{v}}^T(\mathbf{z}_v')\overline{\mathbf{Y}}_{\dot{v}} \ . \tag{8.117}$$

where

$$\mathbf{z}_v' \triangleq \{\dot{o}'(0,t),\ v'(L,t),\ \dot{v}'(L,t),\ V_m'\}^T \ . \tag{8.118}$$

The prefilters are defined as

$$\begin{cases} \dot{o}'(0,t) \triangleq f_{\dot{o}}[\dot{o}(0,t)] \triangleq \dot{o}(0,t)\cdot\frac{2}{0.5\cdot(6+6)} \ . \\ v'(L,t) \triangleq f_v[v(L,t)] \triangleq v(L,t)\cdot\frac{2}{0.5\cdot(0.03+0.03)} \ . \\ \dot{v}'(L,t) \triangleq f_{\dot{v}}[\dot{v}(L,t)] \triangleq \dot{v}(L,t)\cdot\frac{2}{0.5\cdot(0.25+0.25)} \ . \\ V_m' \triangleq f_{V_m}(V_m) \triangleq V_m\cdot\frac{2}{0.5\cdot(0.9+0.9)} \ . \end{cases} \tag{8.119}$$

Both $\mathbf{\Theta}_v$ and $\mathbf{\Theta}_{\dot{v}}$ have 625 free parameters that are trained using adaptive law, eq.(8.34). Use of a difference to approximate the derivative in eq.(8.34) gives

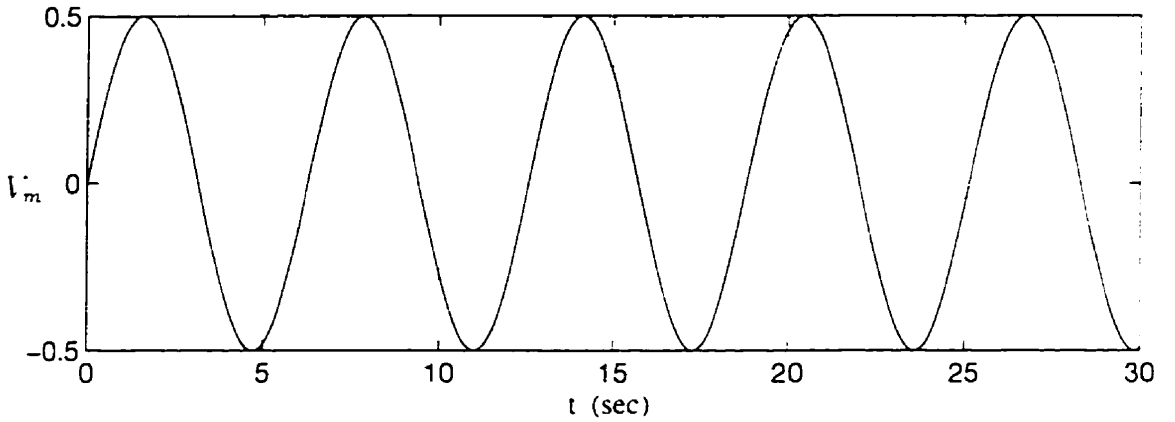
$$\begin{cases} \overline{\mathbf{Y}}_v(kT_v + T_v) = \overline{\mathbf{Y}}_v(kT_v) - T_v[\mathbf{H}_v\mathbf{\Theta}_v e_v + S_v\beta_v\mathbf{H}_v\overline{\mathbf{Y}}_v(kT_v)] \ . \\ \overline{\mathbf{Y}}_{\dot{v}}(kT_v + T_v) = \overline{\mathbf{Y}}_{\dot{v}}(kT_v) - T_v[\mathbf{H}_{\dot{v}}\mathbf{\Theta}_{\dot{v}} e_{\dot{v}} + S_{\dot{v}}\beta_{\dot{v}}\mathbf{H}_{\dot{v}}\overline{\mathbf{Y}}_{\dot{v}}(kT_v)] \ . \end{cases} \tag{8.120}$$

where the time increment, T_v, is set to 0.02 sec, the gain matrices, $\mathbf{H}_v$ and $\mathbf{H}_{\dot{v}}$, are diagonal matrices with all diagonal elements equal to 200, and the bounds of the parameter vectors, M_v and $M_{\dot{v}}$, are set to 10^3. The constants, α_v and $\alpha_{\dot{v}}$, are set to 10.

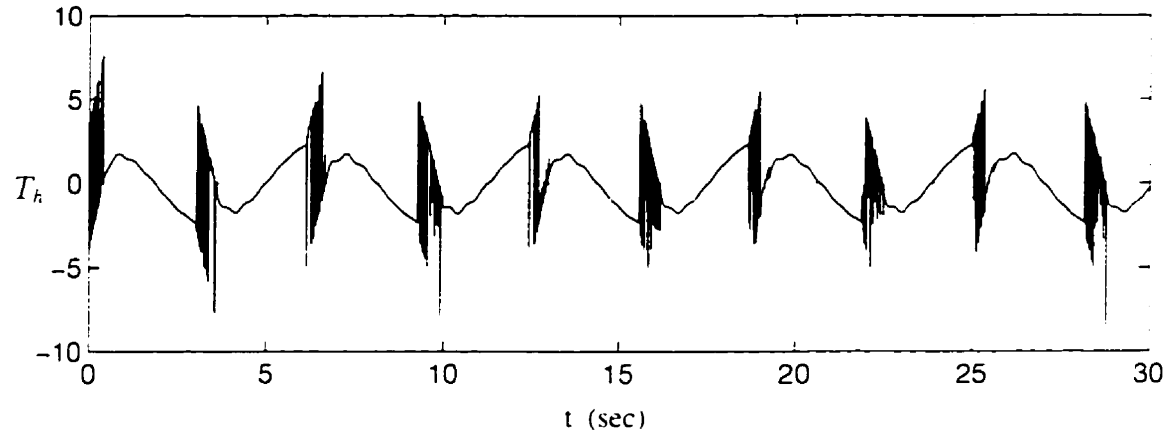
C Training and Testing of DFLS Identifiers

For command input

$$V_m = 0.5\sin(t) \ . \tag{8.121}$$



(a) Command Voltage, $V_m = 0.5\sin(t)$



(b) Torque on the Rotor of the Motor

Figure 8.16: Command Voltage $V_m = 0.5\sin(t)$ and Torque T_h - Example 3

the command voltage, V_m, and the corresponding torque acting on the rotor of the motor, $T_h \triangleq K_T V_m - T_f$, are shown in fig.8.16(a-b). The identification process for identifiers F_ϕ and F_ζ are shown in fig.8.17(a-b), where training ends at $t = 20$ sec.

The identification processes of identifiers F_v and F_i are illustrated in fig.8.18(a-b), where training ends at $t = 40$ sec. The elements of all the parameter vectors, $\overline{\mathbf{Y}}_\phi$, $\overline{\mathbf{Y}}_\zeta$, $\overline{\mathbf{Y}}_v$, and $\overline{\mathbf{Y}}_i$, are initially assigned random values uniformly distributed in (-1, 1). It is seen that all of the four variables have been well identified for the specific input.

Next we trained identifiers using command voltage

$$V_m = A_i \sin(2\pi f_j t) \ . \tag{8.122}$$

which is characterized by the pair $\{A_i,\ f_j\}$. Let

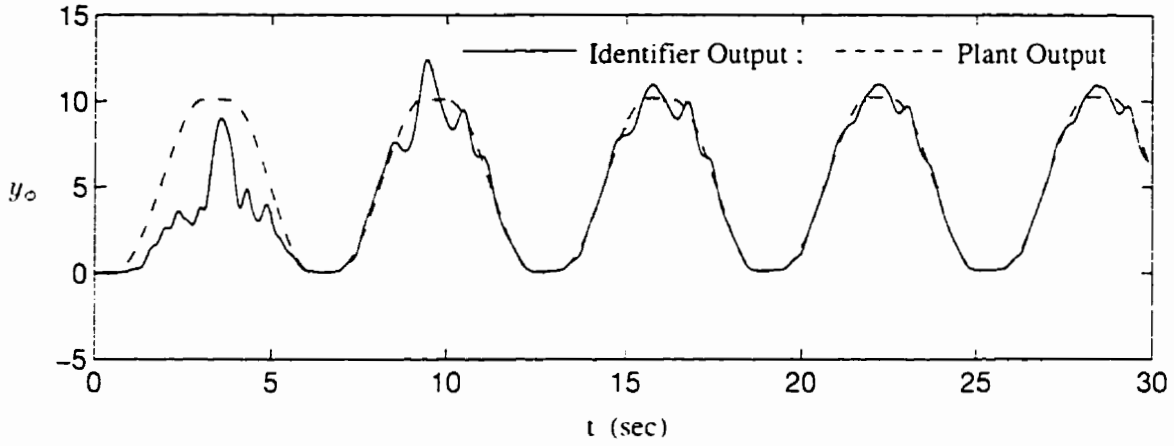
$A_1 = 0.5$ V, $A_2 = 0.3$ V, $A_3 = 0.7$ V.

$f_1 = 0.1$ Hz, $f_2 = 0.2$ Hz, $f_3 = 0.5$ Hz, $f_4 = 1$ Hz, $f_5 = 0.3$ Hz, $f_6 = 0.7$ Hz.

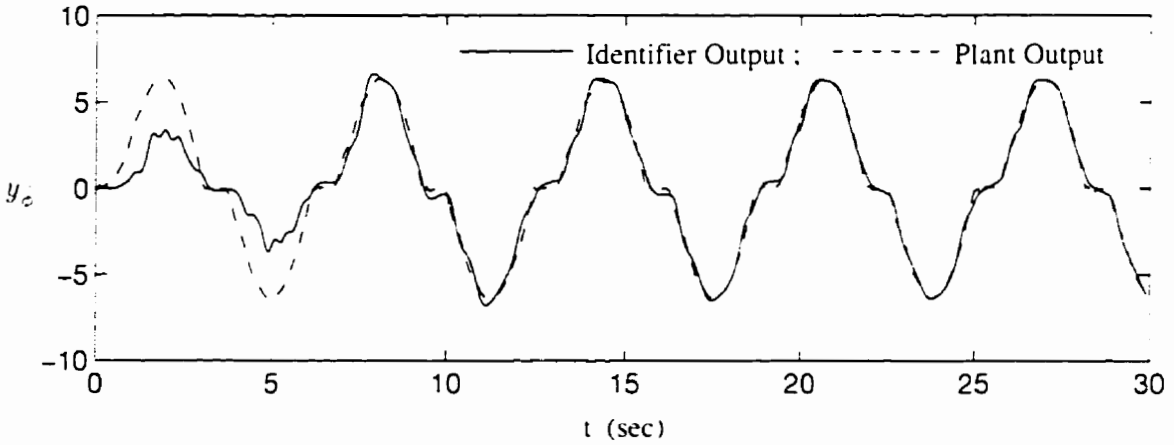
There were eighteen different pairs, $\{A_i,\ f_j\}$, for $i = 1,2,3$, $j = 1,\ldots,6$, which were used to train the DFLS identifiers, and for each pair, training lasted twenty seconds. At the conclusion of training, the trained identifiers were tested against a new command.

$$V_m = 0.3\sin(5t) + 0.4\sin(4t) \ . \tag{8.123}$$

The corresponding torque, $T_i \triangleq K_T V_m - T_f$, is shown in fig.8.19. The outputs of identifiers are shown in fig.8.20(a-b) and fig.8.21(a-b), respectively. The results are quite satisfactory and suggest that the DFLS identifiers and the training algorithm developed in the previous sections can be very effective in robotic applications.



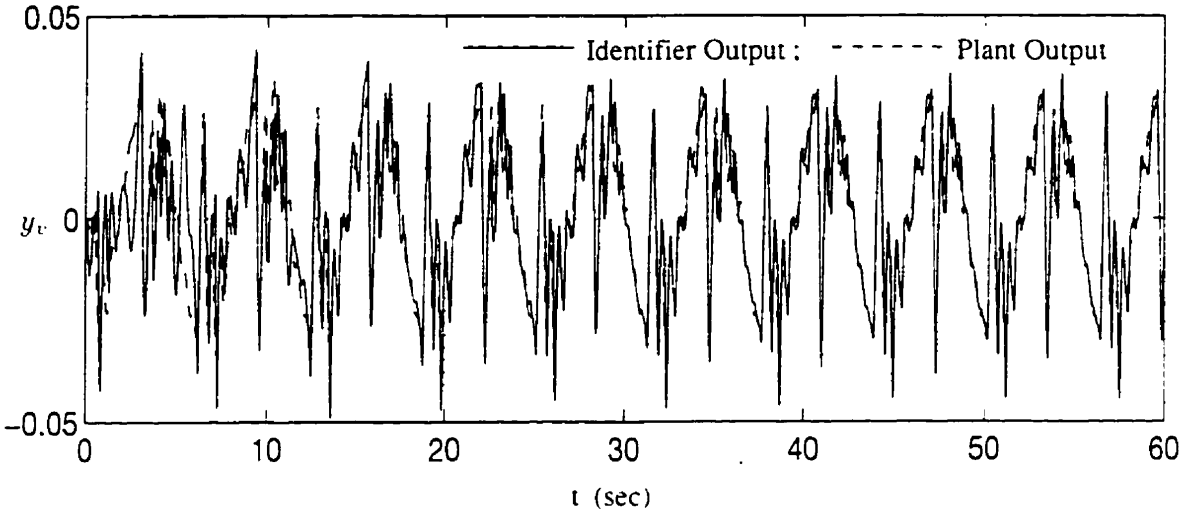
(a) Identification for Angular Displacement $o(0,t)$



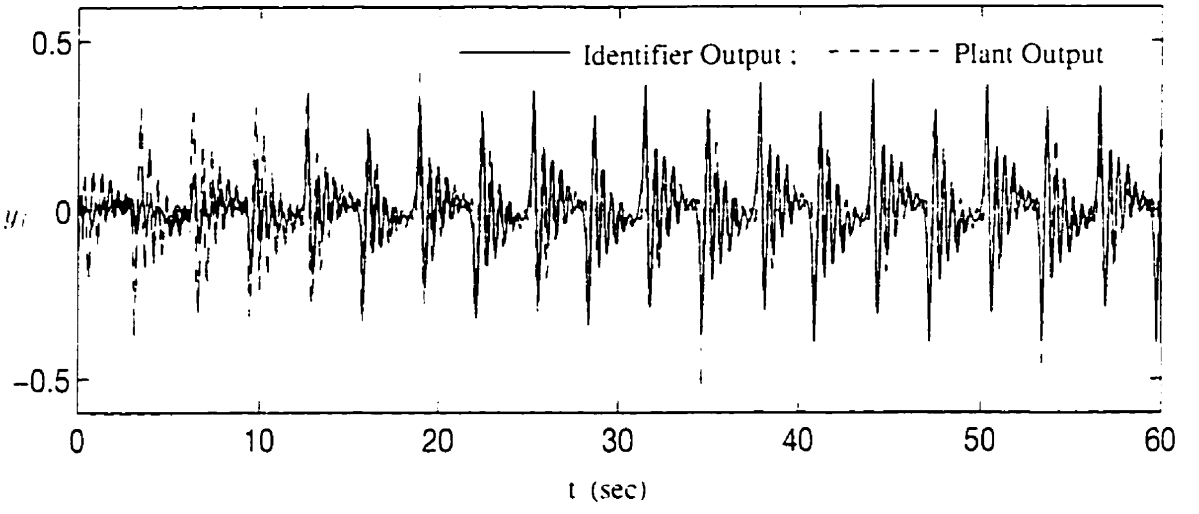
(b) Identification for Angular Velocity $\dot{o}(0,t)$

(Training Terminates at $t = 20$ sec)

Figure 8.17: Identification of Angular Displacement and Velocity for $V_m = 0.5\sin(t)$ - Example 3



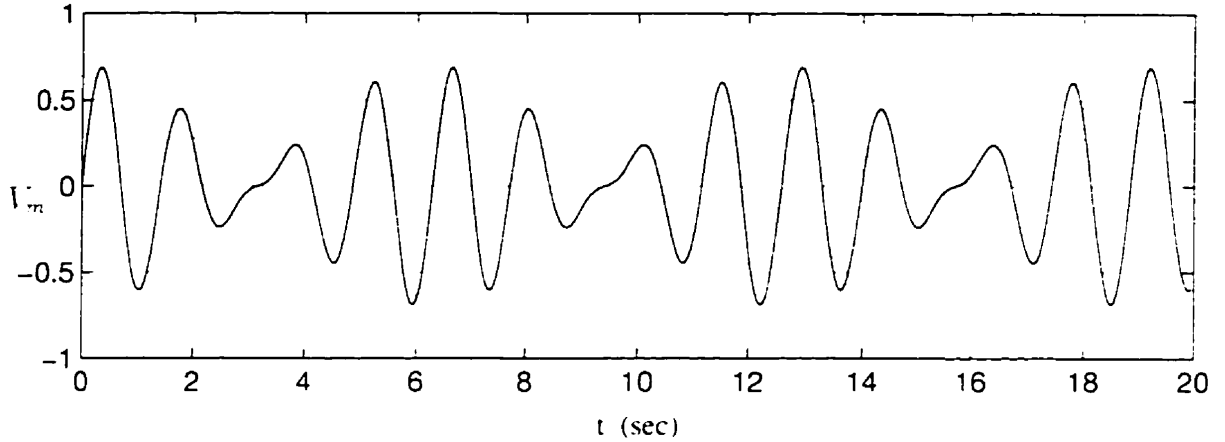
(a) Identification for Vibration Displacement $v(L,t)$



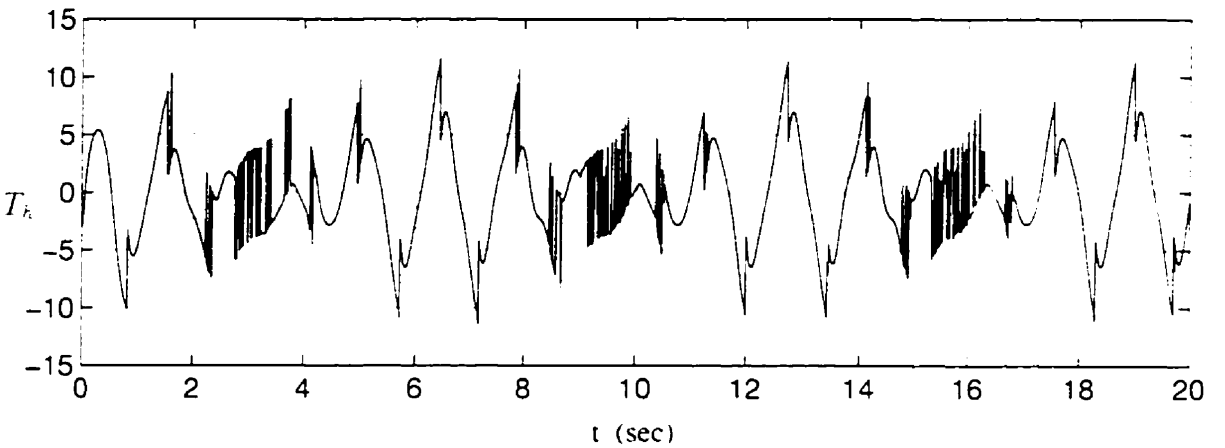
(b) Identification for Vibration Velocity $\dot{v}(L,t)$

(Training Terminates at $t = 40$ sec)

Figure 8.18: Identification of Vibrational Displacement and Velocity at the Tip of the Beam for $V_m = 0.5\sin(t)$ - Example 3

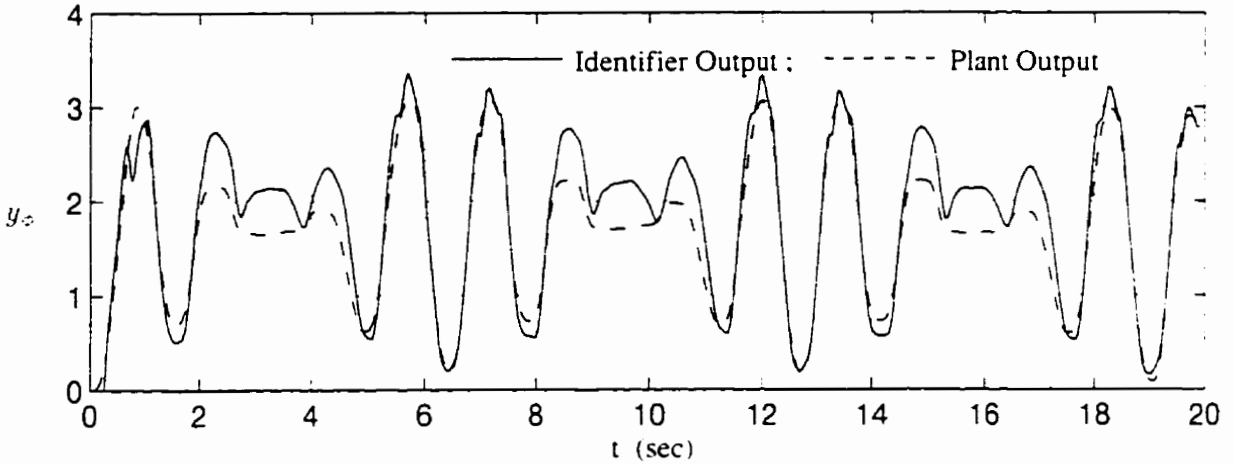


(a) Command Voltage on the Motor. $V_m = 0.3\sin(5t) + 0.4\sin(4t)$

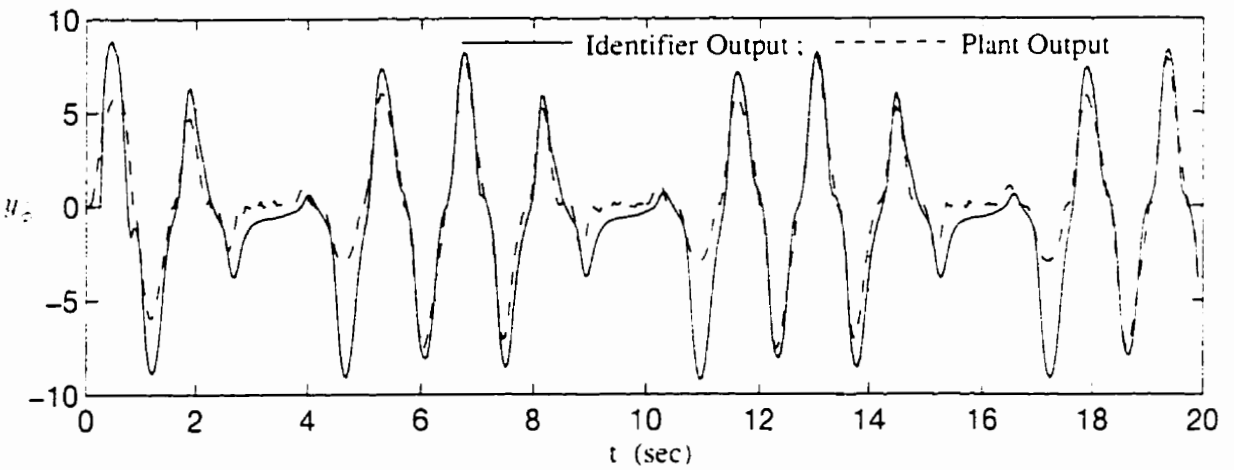


(b) Torque on the Rotor of the Motor

Figure 8.19: Command Voltage V_m and Torque T_h for Test of Trained DFLS Identifier - Example 3

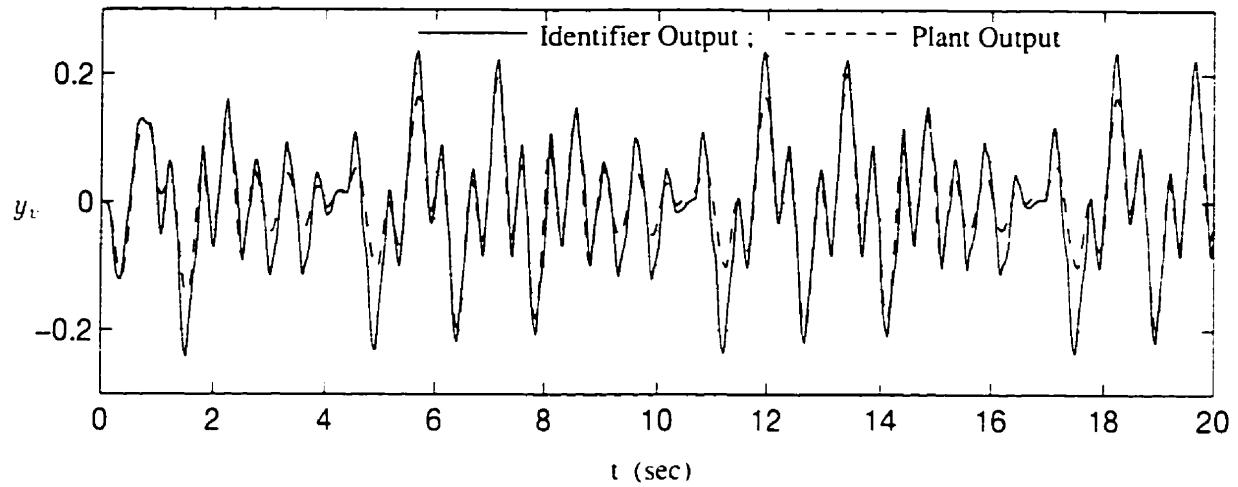


(a) Test of Trained Identifier for Angular Displacement, $\phi(0,t)$

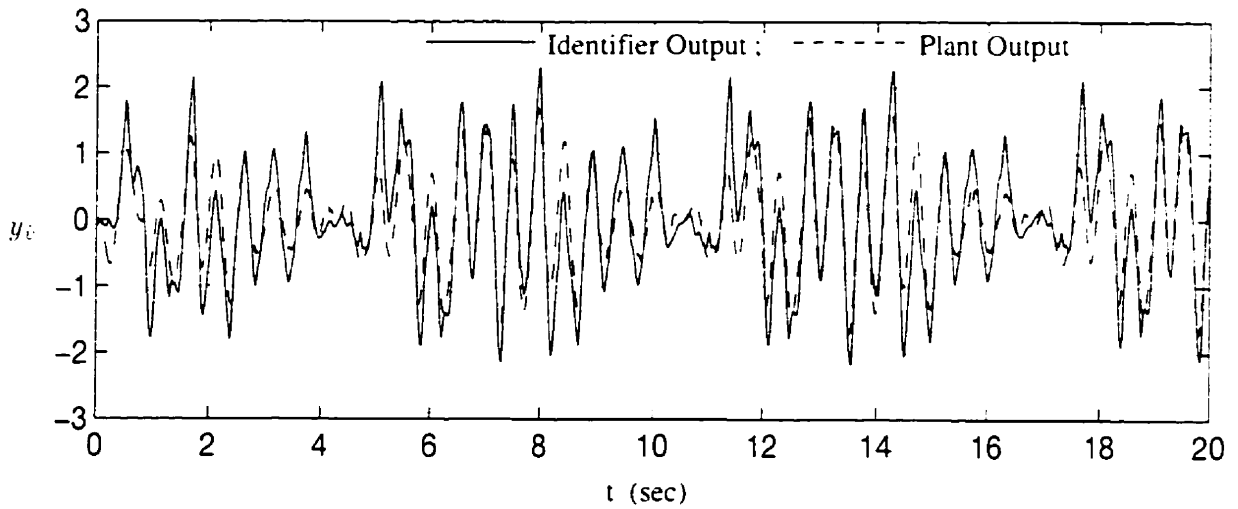


(b) Test of Trained Identifier for Angular Velocity, $\dot{\phi}(0,t)$

Figure 8.20: Test of Trained Identifiers for Angular Displacement and Velocity of the Motor - Example 3



(a) Test of Trained Identifier for Vibration Displacement, $v(L,t)$



(b) Test of Trained Identifier for Vibration Velocity, $\dot{v}(L,t)$

Figure 8.21: Test of Trained Identifiers for Vibrational Displacement and Velocity of the Tip of the Beam - Example 3

8.4.4 Expert Knowledge and Identifier Performance

If some knowledge of system behavior is available *a priori*, it can be integrated into DFLS design and improve its performance. Sometimes even very limited, intuitive knowledge may significantly improve system performance. We demonstrate this point with a simulation experiment.

Consider the flexible link manipulator just described in section 8.4.3. Suppose we had some knowledge of its dynamic behavior. We could now identify the angular velocity of the motor hub, $\dot{o}(0,t)$, using the DFLS identifier, D_2, designed in that section, and incorporate this knowledge of $\dot{o}(0,t)$ into D_2. This knowledge of $\dot{o}(0,t)$ is summarized in Table 8.1 as IF-THEN rules.

Table 8.1: Linguistic Description of Dynamic Behavior of $\dot{o}(0,t)$

IF	[$\dot{o}(0,t)$ is N2]	AND	[$V_m(t)$ is N2]	THEN	[y_2 is N2];
IF	[$\dot{o}(0,t)$ is N2]	AND	[$V_m(t)$ is N1]	THEN	[y_2 is N2];
IF	[$\dot{o}(0,t)$ is N2]	AND	[$V_m(t)$ is ZR]	THEN	[y_2 is N2];
IF	[$\dot{o}(0,t)$ is N1]	AND	[$V_m(t)$ is N2]	THEN	[y_2 is N1];
IF	[$\dot{o}(0,t)$ is N1]	AND	[$V_m(t)$ is N1]	THEN	[y_2 is N1];
IF	[$\dot{o}(0,t)$ is N1]	AND	[$V_m(t)$ is ZR]	THEN	[y_2 is N1];
IF	[$\dot{o}(0,t)$ is ZR]	AND	[$V_m(t)$ is ZR]	THEN	[y_2 is ZR];
IF	[$\dot{o}(0,t)$ is P1]	AND	[$V_m(t)$ is ZR]	THEN	[y_2 is P1];
IF	[$\dot{o}(0,t)$ is P1]	AND	[$V_m(t)$ is P1]	THEN	[y_2 is P1];
IF	[$\dot{o}(0,t)$ is P1]	AND	[$V_m(t)$ is P2]	THEN	[y_2 is P1];
IF	[$\dot{o}(0,t)$ is P2]	AND	[$V_m(t)$ is ZR]	THEN	[y_2 is P2];
IF	[$\dot{o}(0,t)$ is P2]	AND	[$V_m(t)$ is P1]	THEN	[y_2 is P2];
IF	[$\dot{o}(0,t)$ is P2]	AND	[$V_m(t)$ is P2]	THEN	[y_2 is P2].

Initially, those parameters in $\overline{\mathbf{Y}}_{\dot{o}}$ that are relevant to the rules are assigned values according to these rules, and the remaining parameters, i.e., those not affected by the rules are still assigned random numbers uniformly distributed in (-1, 1).

For the new identifier, the training process for a command input,

$$V_m = 0.5\sin(0.2\pi t)\,, \tag{8.124}$$

is shown in fig.8.22, where the dotted line represents the measured values, the dashed line represents identified $\dot{o}(0,t)$ without any initial knowledge, the solid line is the identified $\dot{o}(0,t)$ with expert IF-THEN rules incorporated into $D_{\dot{o}}$ before training starts. The significance of this expert knowledge in performance improvement is clearly demonstrated.

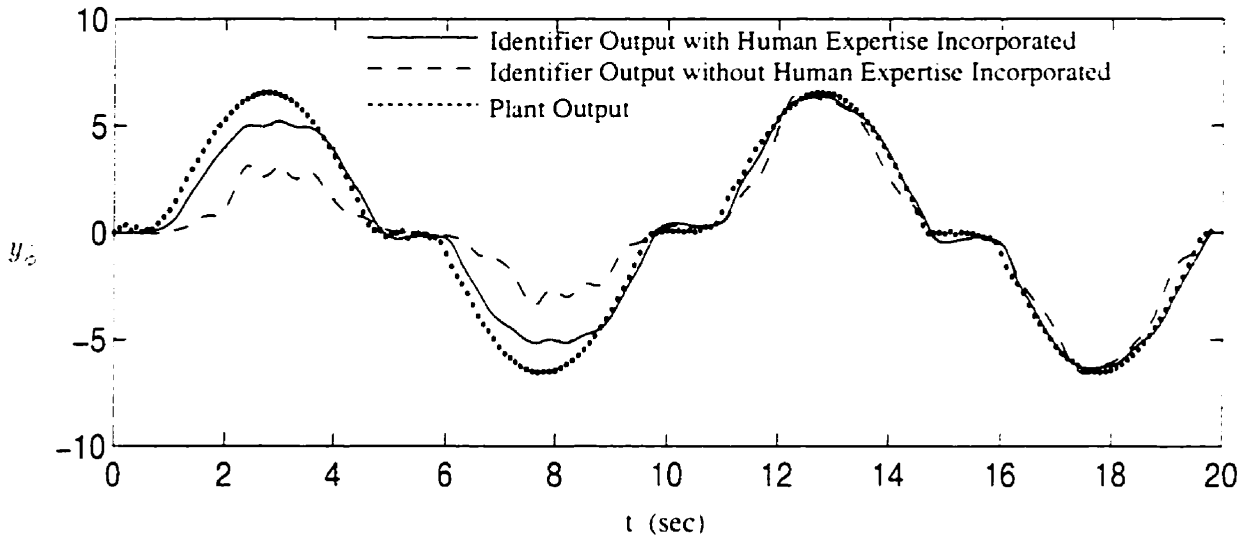


Figure 8.22: Training of the Identifier for Angular Velocity of the Motor, $\dot{o}(0,t)$, with and without Expert Knowledge Incorporated

8.5 Experimental Demonstration – Identification of a Mechanical Manipulator

To further demonstrate the properties and effectiveness of the DFLS identification algorithm, DFLS identifiers are applied to identify the mechanical single link manipulator described in Appendix B. This mechanical system is characterized by a very flexible link, significant nonlinear joint friction and an unknown internal motor speed control loop.

The angular displacement of the motor hub, $o(0,t)$, and the tip vibrational displacement of the link, $v(L,t)$, are to be identified. To simplify the notation, let

$$\begin{cases} o_m(t) \triangleq o(0,t) , \\ v_L(t) \triangleq v(L,t) , \end{cases} \tag{8.125}$$

Two DFLS identifiers, D_o and D_v, are used for variables, $o(0,t)$ and $v(L,t)$, respectively, i.e.,

$$\begin{cases} y_o = D_o(o_m, \dot{o}_m, V_m) , \\ y_v = D_v(\dot{o}_m, v_L, \dot{v}_L, V_m) , \end{cases} \tag{8.126}$$

This is illustrated in fig.8.23.

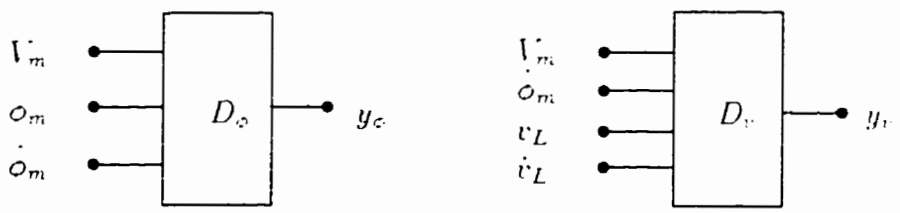


Figure 8.23: Inputs-Outputs of DFLS identifiers - Identification Experiment

The identification errors are denoted as

$$\begin{cases} \epsilon_o \triangleq y_o - o(0,t) , \\ \epsilon_v \triangleq y_v - v(L,t) , \end{cases} \tag{8.127}$$

Each universe of discourse is partitioned into five primary fuzzy sets, namely,

$$\{N2,\ N1,\ ZR,\ P1,\ P2\},$$

whose membership functions are the same as those defined in eqs.(8.74-8.76) and illustrated in fig.8.3.

The gain matrices, $\mathbf{H}_\phi$ and $\mathbf{H}_v$, are taken to be diagonal matrices with all diagonal elements equal to 40. The bounds of parameter vectors, M_ϕ and M_v, are set to 10^3, and the constants, α_ϕ and α_v, to 10. These parameters are selected quite arbitrarily without a great deal of effort and thus may not be optimal ones, which poses a more stringent condition for our training algorithm. Experimental data are obtained at a sampling rate of 40 Hz.

The identification process for command voltage

$$V_m = 1.2\sin(0.2\pi t)\ (V), \tag{8.128}$$

is shown in fig.8.24, with fig.8.24(a) the command voltage, 8.24(b) the angular displacement, and 8.24(c) the tip vibrational displacement. The parameter vectors to be trained, $\overline{\mathbf{Y}}_\phi$ and $\overline{\mathbf{Y}}_v$, are both initially assigned random numbers uniformly distributed in (-1, 1).

The training process for ϕ_m converges very fast, as seen in fig.8.24(b). Figure 8.24(c) shows the training process for the tip vibrational displacement, v_L, which is the 14th round of training (i.e., the identifier is repeatedly trained 14 times using the same batch of data). It converges well although it requires more effort than does the hub angle, which is not surprising since the dynamics of the tip vibration is much more complex than that of the hub angle.

For this low command voltage, the motor rotates at very low speed and stiction has a very significant effect on system behavior and makes system dynamics highly nonlinear[36], as observed from fig.8.24.

Figure 8.25 illustrates the training process for command input

$$V_m = 2\sin(0.6\pi t)(V) \ . \tag{8.129}$$

The trained parameter vectors from fig.8.24 are used as the starting points for this training. Figure 8.25(c) shows the 2nd round of training. Again, good convergence is evident for both variables.

Figure 8.26 shows the training process for a complex command input pattern, shown in fig.8.26(a), which is generated by a human controlled joystick superposed with some disturbances. Thus the input is quite arbitrary and nonrepeatable. The initial parameter vectors are obtained by training the identifier with command inputs

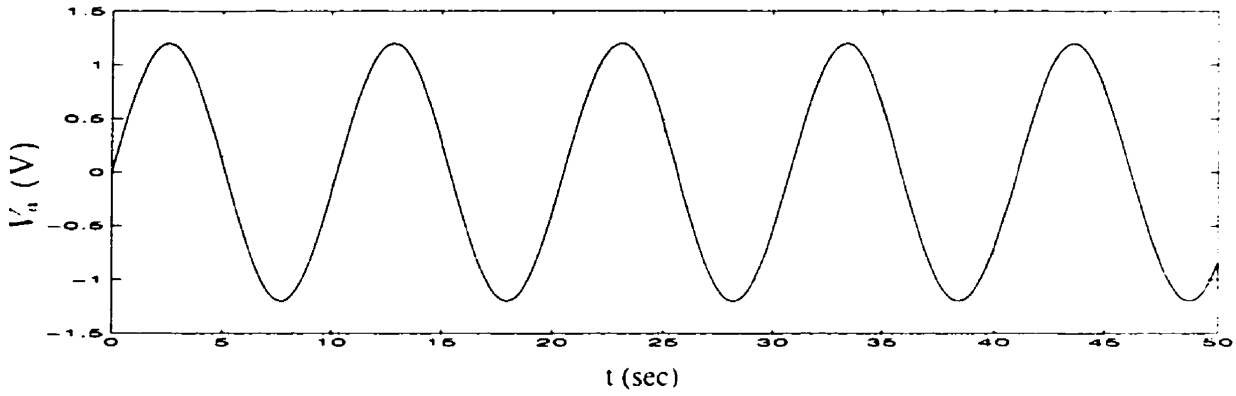
$$V_m = A_i \sin(2\pi f_j t)\ (V) \ , \tag{8.130}$$

where

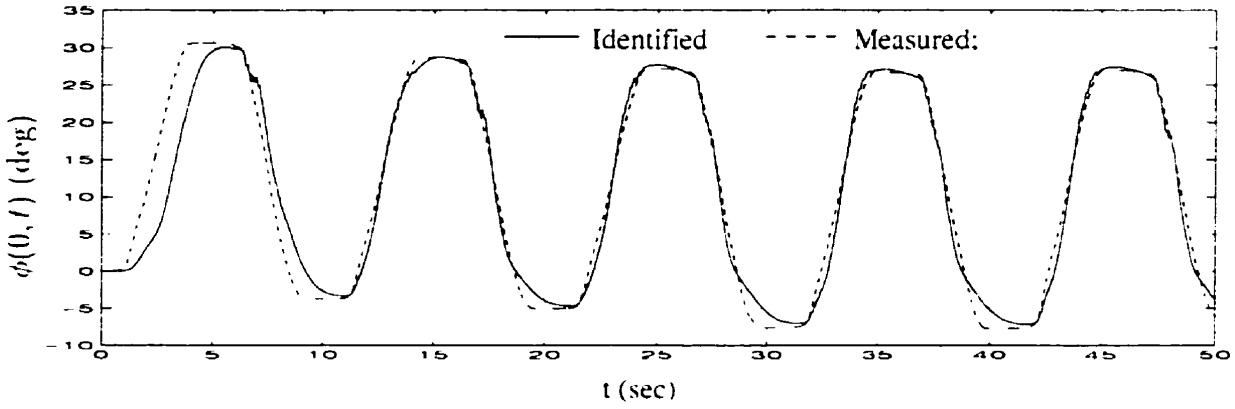
$$\begin{cases} A_i & \in \ \{1.2,\ 1.5,\ 2,\ 2.5,\ 3,\ 3.5,\ 4,\ 4.5,\ 5,\ 5.5\} \ . \\ f_j & \in \ \{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9\} \ . \end{cases} \tag{8.131}$$

Each input pattern $\{A_i,\ f_j\}$ lasts fifty seconds, and the initial elements of $\overline{\mathbf{Y}}_k$, $k = 1,2$, for the first pattern $\{A_1 = 1.2,\ f_1 = 0.1\}$, are random numbers uniformly distributed in $(-1,\ 1)$. It is clear that even for such complex inputs, the identification process converges reasonably well.

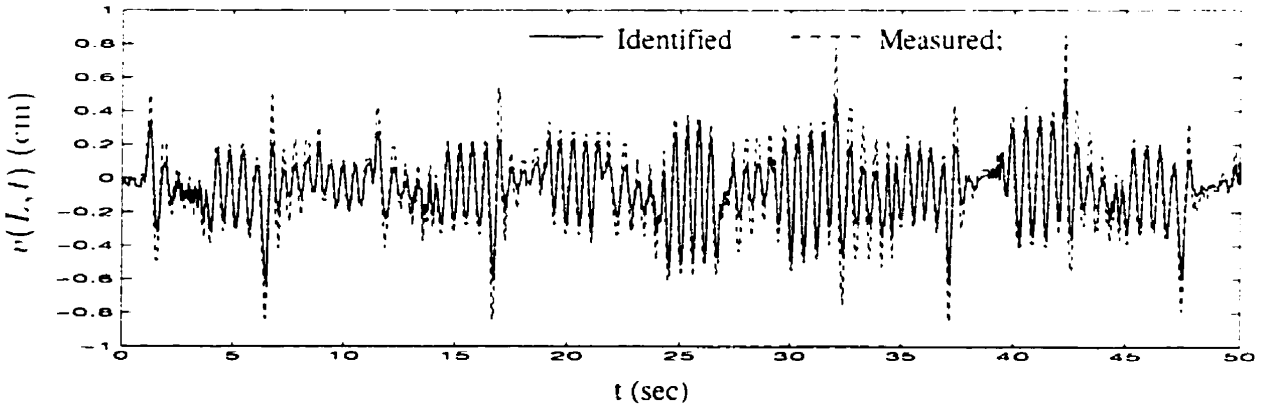
After training with an additional twenty five complex input patterns, each of which lasted thirty seconds, the trained identifiers were tested by predicting the responses of $o(t)$ and $v(t)$ for the inputs shown in fig.8.27(a), which is quite complex (and was used in the training process). Figure 8.27(b-c) shows the responses for $o(0,t)$ and $v(L,t)$ predicted by the identifiers. The results are quite satisfactory and suggest that the proposed DFLS identifiers and the training algorithm can be very effective in practical applications.



(a) Command Input: $V_a = 1.2\sin(0.2\pi t)$

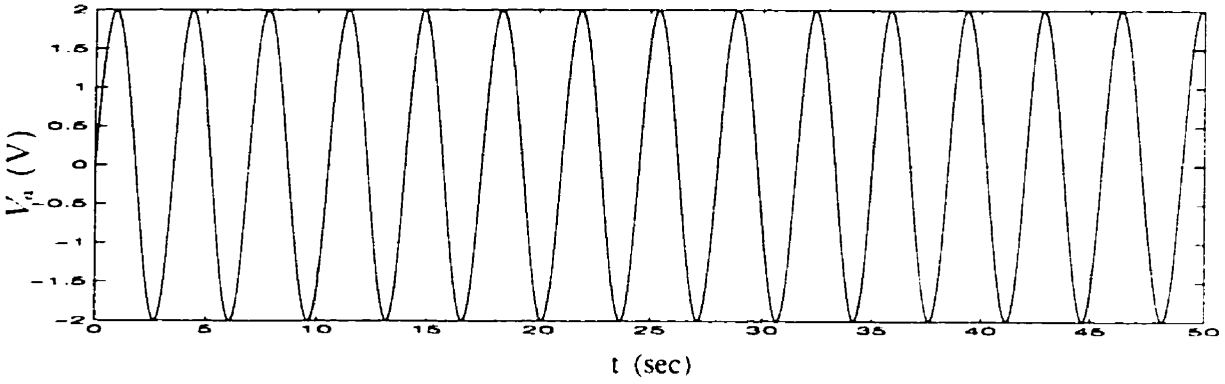


(b) Angular Displacement

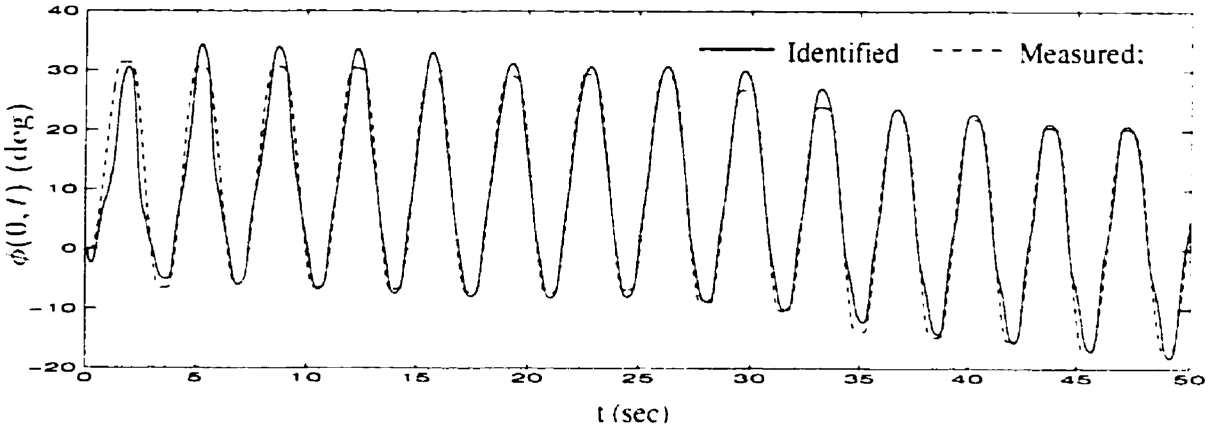


(c) Vibrational Displacement of the Beam Tip

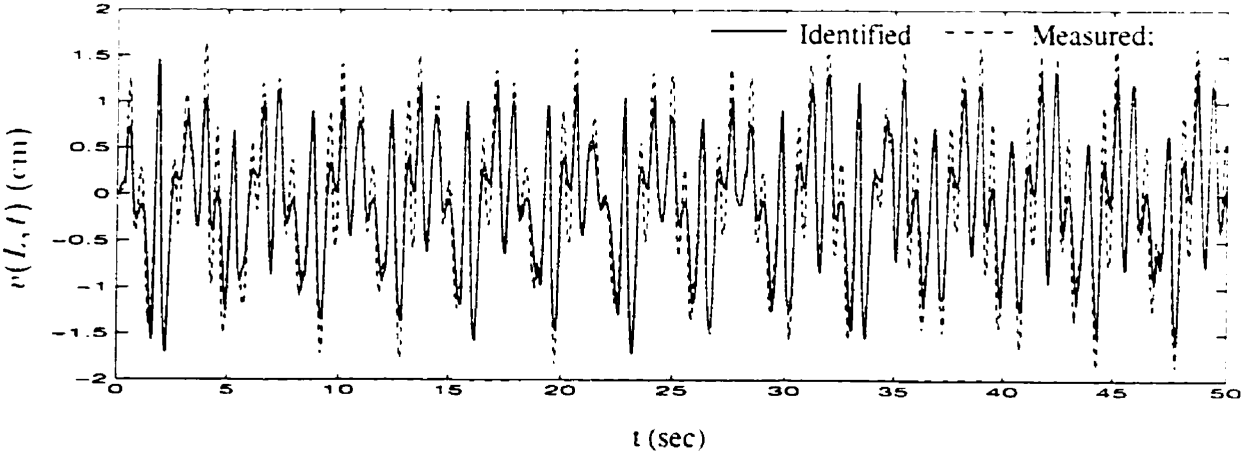
Figure 8.24: Training Process for Command Input $V_m = 1.2\sin(0.2\pi t)$ - Identification Experiment



(a) Command Voltage $V_a = 2\sin(0.6\pi t)$

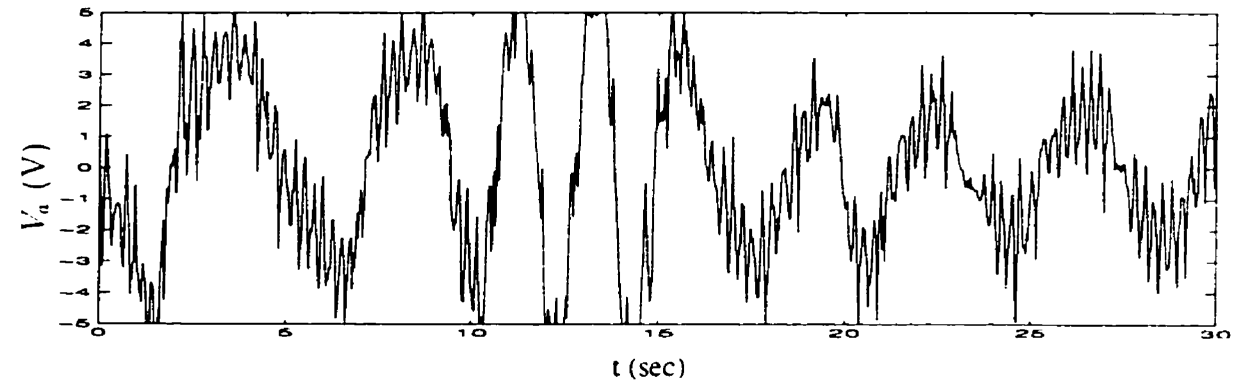


(b) Angular Displacement of Motor Hub

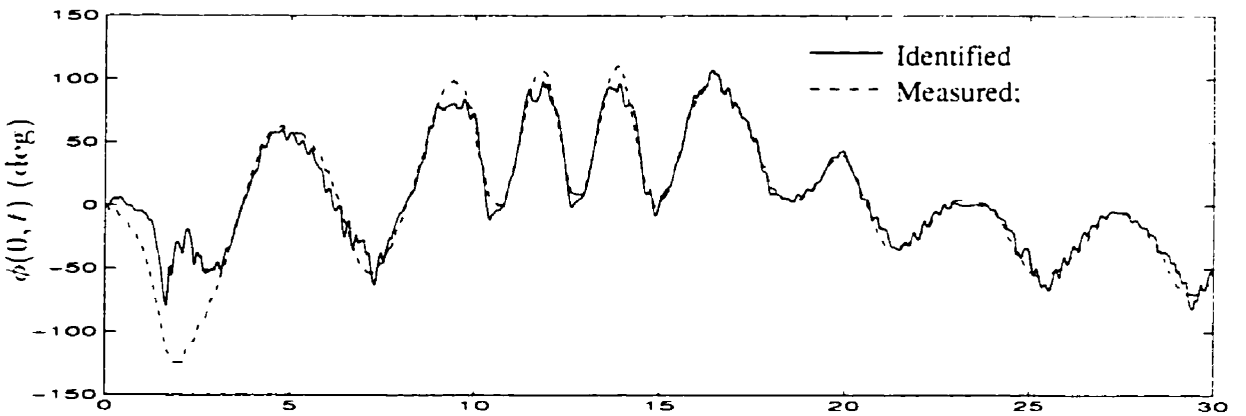


(c) Vibrational Displacement of the Beam Tip

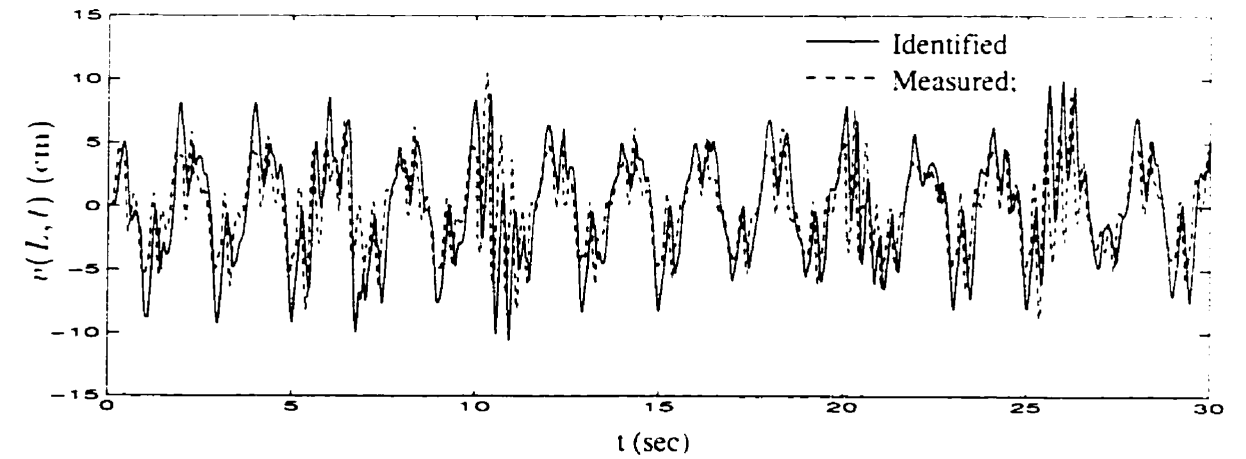
Figure 8.25: Training Process for Command Input $V_m = 2\sin(0.6\pi t)$ - Identification Experiment



(a) Command Voltage

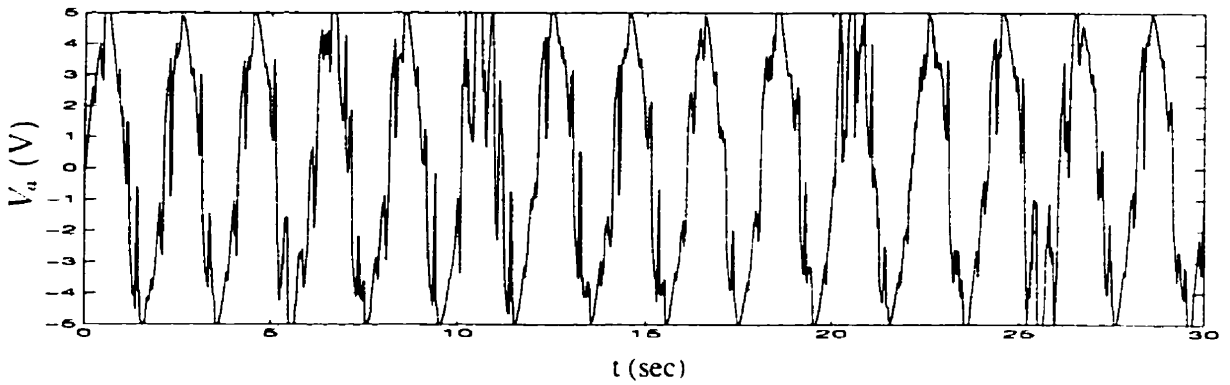


(b) Angular Displacement of Motor Hub

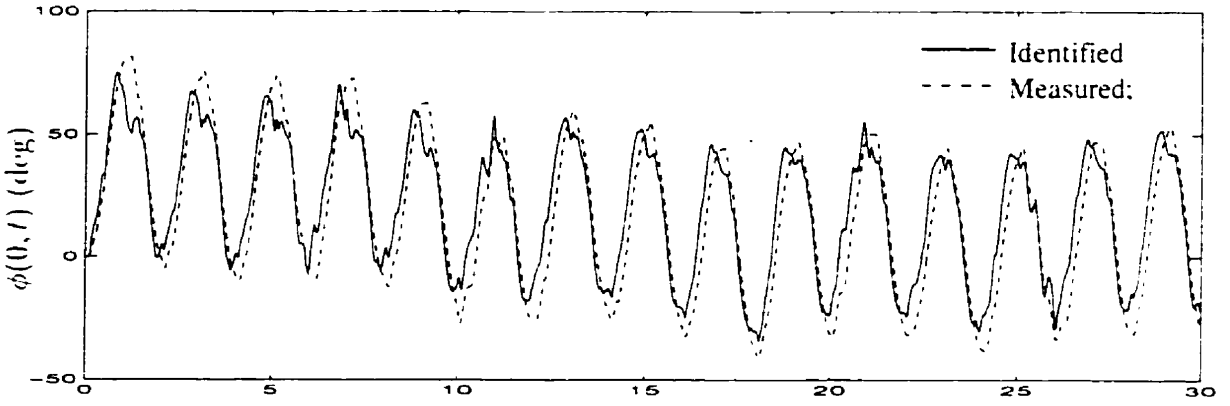


(c) Vibrational Displacement of the Beam Tip

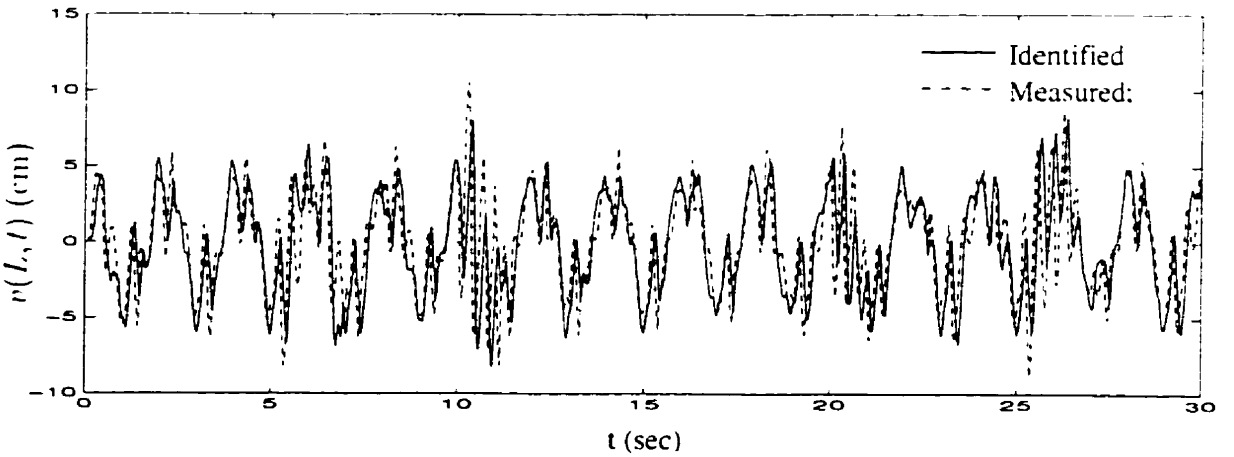
Figure 8.26: Training Process for Complex Command Input - Identification Experiment



(a) Command Input



(b) Angular Displacement of Motor Hub



(c) Vibrational Displacement of the Beam Tip

Figure 8.27: Test of Identifiers for Complex Command Input - Identification Experiment

8.6 Concluding Remarks

In addition to the remarks made in section 8.3.4, we have the following conclusions and observations.

1. A novel fuzzy logic system structure, which is characterized by its dynamic aspect, is proposed here, and its universal approximation property proven. We believe that the new dynamic fuzzy logic system, DFLS, is more naturally integrated into many dynamic systems and makes better use of the intrinsic dynamics than does the conventional static version.

2. A DFLS based stable adaptive identification algorithm is developed here, which enables the DFLS identifiers to identify a large class of nonlinear dynamic systems reliably.

3. The DFLS based identification algorithm has been applied to identify a variety of nonlinear system, and the design procedures of DFLS identifiers have been demonstrated in detail. In all of the applications, satisfactory results were obtained and the effectiveness of DFLS identifiers in dealing with complex, nonlinear systems is clearly demonstrated.

4. In the *Chaotic Glycolytic Oscillator* example, section 8.4.2, the identification results of the DFLS identifiers are compared with those of static FLS identifiers in published literature, and the former is clearly better. This example offers an evidence to support one of our assertion that DFLS can have performance superior to that of a static FLS, although more in-depth comparisons between DFLS and static FLS are yet to be done. This is left for future investigation.

5. In most of our applications, it was assumed that there is no human expertise available *a priori*, which poses a more stringent condition on DFLS identifiers, but satisfactory results can still be achieved nevertheless.

If there is expert knowledge available *a priori* in form of IF-THEN rules, even of a very limited nature, it can be explicitly incorporated into identifiers and significantly improve system performance. This is demonstrated through a simulation experiment in section 8.4.4.

6. In this work, the off line designed parameters, in view of eqs.(8.23) and (8.34), include α_k, $\mathbf{H}_k$, $M_{\overline{\mathbf{Y}}_k}$ and $\mathbf{\Theta}_k$, as well as prefilter, f_k, and time increment, T. These were all determined by trial and error. Explicit procedures for determining the optimal values of these parameters have yet to be developed.

   It should be pointed out that in the illustrative applications, these parameters are selected without a great deal of effort and they are thus by no means considered to be optimal, which also poses more stringent demands on the DFLS identification algorithm.

7. The conditions for *adequate training* of a DFLS are not discussed here, which itself is an interesting research topic and involves issues such as persistent input excitation, the choices of fuzzy logic system structure and the off-line design parameters, and so on. This is a subject for future investigation.

8. Problems associated with system and measurement noises are not discussed in this work. This is another subject for future investigation.

Chapter 9

On Adaptive Control of Nonlinear Systems – DFLS Approach

9.1 Introduction

In chapter 6, we presented an empirical fuzzy logic controller design process which is solely based on expert knowledge. The resulting FLC was shown to work well if there is adequate expertise available and the control tasks are not complex. The main problem with this approach is that it does not have any theoretical guarantee of stability or consistent performance in the presence of large uncertainties and unknown variations in system behavior. In addition, if there is not adequate expert knowledge available, or the control tasks are too complex, this approach will be very difficult to effectively implement. Therefore, there is a need to develop a systematic design and analysis approach for fuzzy logic control systems, such that the resulting fuzzy logic control systems are adaptive in the sense that they can not only be capable of incorporating linguistic fuzzy information from human experts, but also maintain system stability and consistent performance in the presence of system uncertainties and variations. Since the latter part of the objective generally coincides with the basic objective of adaptive control, which is to maintain consis-

tent performance in the presence of variations of system parameters, the design and analysis tools from the realm of adaptive control may be useful in the development of adaptive fuzzy logic systems.

Pioneering work of this kind was reported in [182] and [154, 184]. The former developed a direct adaptive fuzzy logic control scheme where fuzzy logic systems are used as controllers and their parameters directly adjusted to reduce some norm of the output error between the plant and the reference model. Linguistic IF-THEN rules can be directly incorporated into controllers of this kind. The latter developed an indirect adaptive fuzzy logic control scheme where fuzzy logic systems are used to model the plant whose parameters are estimated, and the controller is chosen assuming that the estimated parameters represent the true values of the plant parameters. The fuzzy IF-THEN rules describing the plant can be directly incorporated into the indirect adaptive fuzzy controllers.

However, in these works the fuzzy logic systems involved are static in nature. In this chapter, the dynamic fuzzy logic system described in chapter 8 is used to develop a stable indirect adaptive fuzzy logic control scheme, following the Lyapunov synthesis approach. The resulting DFLS based adaptive controller will be seen to have certain advantages over static FLS based adaptive controllers.

To demonstrate the application and properties of the DFLS adaptive controller simulation examples are presented here in which it is used to control various nonlinear systems. To further justify the properties and illustrate its effectiveness, the DFLS adaptive controller is also applied to control the trajectory (at the tip of the link) of a single link mechanical manipulator, which is characterized by a very flexible link, significant nonlinear joint friction and an unknown internal control loop, as described in the previous chapter.

9.2 Preliminaries and Objective

9.2.1 Dynamic System

Consider the following class of nonlinear system:

$$x^{(N)} = f(\mathbf{x}) + g(\mathbf{x})u \ . \tag{9.1}$$

where $u \in R$ and $x \in R$ are the input and output of the system, respectively. $\mathbf{x} \triangleq \{x, \dot{x}, \ldots, x^{(N-1)}\}^T \in R^N$ is the state vector which is assumed to be available for measurement, and $x^{(N)}$ represents the Nth order derivative of x with respect to time. $f : R^N \rightarrow R$ and $g : R^N \rightarrow R$ are unknown continuous nonlinear functions defined in certain controllability region $\mathcal{X} \subset R^N$. Let

$$\begin{cases} x_1 & \triangleq & x \ , \\ x_2 & \triangleq & \dot{x} \ , \\ \vdots & & \\ x_N & \triangleq & x^{(N-1)} \ . \end{cases} \tag{9.2}$$

Then $\mathbf{x} = \{x_1, \ldots, x_N\}^T$, and eq.(9.1) can be written in state space form as,

$$\begin{cases} \dot{x}_1 & = & x_2 \ , \\ \vdots & & \\ \dot{x}_{N-1} & = & x_N \ , \\ \dot{x}_N & = & f(\mathbf{x}) + g(\mathbf{x})u \ . \end{cases} \tag{9.3}$$

For the system to be controllable, we impose mild restrictions on the function g.

Assumption 9.1 $0 < \varepsilon_g \leq |\ g(\mathbf{x})\ | \leq M_g, \ \forall\ \mathbf{x} \in \mathcal{X}$ *and* $0 \leq t < \infty$, *where* M_g *is a* large *positive constant and* ε_g *is a* small *positive constant. The sign of* $g(\mathbf{x})$ *is also known.*

9.2.2 DFLS Identifier

Consider the DFLS of eq.(8.5). i.e.

$$\dot{y} = -\alpha y + \mathbf{\Theta}^T(\mathbf{x}, u)\overline{\mathbf{Y}} \ . \tag{9.4}$$

where $\mathbf{\Theta}$ and $\overline{\mathbf{Y}}$ are as defined in eqs.(8.3-8.4). i.e..

$$\begin{cases} \mathbf{\Theta}(\mathbf{z}) \ \hat{=} \ \{\theta_1(\mathbf{z}), \ldots, \theta_I(\mathbf{z})\}^T \ . \\ \overline{\mathbf{Y}} \ \hat{=} \ \{\overline{y}_1, \ldots, \overline{y}_I\}^T \ . \end{cases} \tag{9.5}$$

and $\theta_i(\mathbf{z})$ is given by eq.(4.43) as.

$$\theta_i(\mathbf{z}) \hat{=} \frac{\prod_{k=1}^{N+1} \exp[-\frac{1}{2}(\frac{z_k - \overline{z}_k^i}{\sigma_k^i})^2]}{\sum_{i=1}^{I} \prod_{k=1}^{N+1} \exp[-\frac{1}{2}(\frac{z_k - \overline{z}_k^i}{\sigma_k^i})^2]} \ . \tag{9.6}$$

where

$$\mathbf{z} \hat{=} \{z_1, \ldots, z_{N+1}\}^T \ \hat{=} \ \{x_1, \ldots, x_N, u\}^T \ . \tag{9.7}$$

This DFLS will be used as an on-line identifier for the unknown state x_N, where parameters α and $\mathbf{\Theta}$ are to be defined in an off-line design process, leaving only $\overline{\mathbf{Y}}$ as a free parameter vector and which is to be adaptively tuned on-line.

The expression for state variable $\dot{x}_N$ in eq.(9.3) is now rewritten as

$$\dot{x}_N = -\alpha x_N + \mathbf{\Theta}^T(\mathbf{x}, u)\overline{\mathbf{Y}} - r(\mathbf{x}, u, \mathbf{\Theta}, \overline{\mathbf{Y}}) \ . \tag{9.8}$$

where

$$r(\mathbf{x}, u, \mathbf{\Theta}, \overline{\mathbf{Y}}) \hat{=} \mathbf{\Theta}^T(\mathbf{x}, u)\overline{\mathbf{Y}} - \alpha x_N - f(\mathbf{x}) - g(\mathbf{x})u \ . \tag{9.9}$$

which may be considered the static modeling error of the DFLS identifier. By Lemma 8.1, for bounded inputs and states there exists an optimal parameter vector, $\overline{\mathbf{Y}}^*$, which minimizes the static modeling error, $r(\mathbf{x}, u, \mathbf{\Theta}, \overline{\mathbf{Y}}^*)$.

$$\overline{\mathbf{Y}}^* \hat{=} \{\overline{\mathbf{Y}}' \mid \min_{\overline{\mathbf{Y}}'}\{\sup_{\{\mathbf{x},u\}\in\mathcal{Z}} \mid r(\mathbf{x}, u, \mathbf{\Theta}, \overline{\mathbf{Y}}') \mid\}\} \ . \tag{9.10}$$

where

$$\overline{\mathbf{Y}}' \triangleq \{\overline{\mathbf{Y}} : \| \overline{\mathbf{Y}} \| \leq M_{\overline{\mathbf{Y}}}\} . \tag{9.11}$$

for which

$$\sup_{\{\mathbf{x},u\}\in\mathcal{Z}} | r(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}') | \leq M_r . \tag{9.12}$$

$\mathcal{Z} \subset R^{N+1}$ is a compact set. $M_{\overline{\mathbf{Y}}}$ and M_r are positive constants that bound the norms of $\overline{\mathbf{Y}}$ and $r(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}')$, respectively.

Here, we know that such an optimal parameter vector exists for bounded inputs and states, but we have no guidance on how to find it. In the following sections, we first develop a control law which ensures the boundedness of the system input and system states, then specify an adaptive law for $\overline{\mathbf{Y}}$, such that $\overline{\mathbf{Y}}^*$ can be satisfactorily approximated.

Replacing $\overline{\mathbf{Y}}$ by $\overline{\mathbf{Y}}^*$ in eq.(9.8) results in

$$\dot{x}_N = -a x_N + \boldsymbol{\Theta}^T(\mathbf{x}, u)\overline{\mathbf{Y}}^* - r(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) . \tag{9.13}$$

Subtracting eq.(9.13) from eq.(9.4) yields

$$\dot{\xi} = -a\xi + \boldsymbol{\Theta}^T(\mathbf{x}, u)\boldsymbol{\Delta}_{\overline{\mathbf{Y}}} + r(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) . \tag{9.14}$$

where ξ is the identification error for the state variable x_N.

$$\xi \triangleq y - x_N . \tag{9.15}$$

and the vector $\boldsymbol{\Delta}_{\overline{\mathbf{Y}}}$ is the parameter estimation error.

$$\boldsymbol{\Delta}_{\overline{\mathbf{Y}}} \triangleq \overline{\mathbf{Y}} - \overline{\mathbf{Y}}^* . \tag{9.16}$$

9.2.3 Objective

Let q be a desired trajectory of system output, x, and denote

$$\begin{cases} q_1 & \triangleq & q \ , \\ q_2 & \triangleq & \dot{q} \ , \\ \vdots & & \\ q_N & \triangleq & q^{(N-1)} \ . \end{cases} \tag{9.17}$$

Let ϵ be the difference between x and q, i.e.,

$$\epsilon \triangleq x - q \ . \tag{9.18}$$

Denote

$$\begin{cases} \epsilon_1 & \triangleq & \epsilon \ , \\ \epsilon_2 & \triangleq & \dot{\epsilon} \ , \\ \vdots & & \\ \epsilon_N & \triangleq & \epsilon^{(N-1)} \ . \end{cases} \tag{9.19}$$

Our objective is to develop DFLS based stable adaptive control system, such that the plant output, $x(t)$, follows the predefined bounded reference trajectory, $q(t)$, under the constraint that all quantities involved must be bounded.

Specifically, we will develop control law for plant input, u, adaptive law for DFLS free parameter vector, $\overline{\mathbf{Y}}$, and adaptive law to estimate the unknown function, g, such that the closed system is stable in the sense that

(i) the tracking error, ϵ, and the identification error, ξ, are bounded and is to be as small as possible, ideally converging to zero;

(ii) the DFLS identifier parameter vector, $\overline{\mathbf{Y}}$, is bounded;

(iii) the system input and system states are bounded;

9.3 Control Law of the System

To make sure that the system states have the desired tracking performance and are always bounded, we specify a control law consisting of two components, u_c and u_s.

$$u \triangleq u_c - S_u u_s \ . \tag{9.20}$$

where

$$S_u \triangleq \begin{cases} 0 \ , & if \quad \| \mathbf{e} \| < M_{\mathbf{e}} \ . \\ 1 \ , & if \quad \| \mathbf{e} \| \geq M_{\mathbf{e}} \ . \end{cases} \tag{9.21}$$

$M_{\mathbf{e}}$ is a finite positive design parameter that bounds $\| \mathbf{e} \|$. The first component, u_c, is the usual control term for tracking performance when the system states are within a desired bound.

$$u_c \triangleq \frac{1}{\hat{g}} \left[\dot{q}_N - \alpha q_N - \boldsymbol{\Theta}^T(\mathbf{x}, 0)\overline{\mathbf{Y}} - \sum_{k=1}^{N-1} \alpha_k e_k \right] . \tag{9.22}$$

where $\alpha_1, \ldots, \alpha_{N-1}$ are design parameters yet to be determined, $\hat{g}$ is the estimated value of the unknown function $g(\mathbf{x})$, and

$$\boldsymbol{\Theta}(\mathbf{x}, 0) \triangleq \boldsymbol{\Theta}(\mathbf{x}, u) \big|_{u \equiv 0} \ . \tag{9.23}$$

The second component, u_s, is a supervisory control term that restrains the system states from drifting beyond a desired bound, and is only activated when system states actually reach the bound. This term follows the idea presented in [183]. Here, it is derived as

$$\begin{aligned} u_s(t) \ \triangleq \ sgn(\mathbf{B}^T \mathbf{P} \mathbf{e}) \cdot sgn(g) \cdot \frac{1}{\varepsilon_g} \cdot \big[\alpha \, | \, x_N \, | + | \, \boldsymbol{\Theta}^T(\mathbf{x}, 0)\overline{\mathbf{Y}} \, | + M_f \\ + (M_g + | \, \hat{g} \, |) \cdot | \, u_c \, | \big] \ . \end{aligned} \tag{9.24}$$

where M_f is a large positive constant that bounds $f(\mathbf{x})$, that is, $\forall \, \mathbf{x} \in \mathcal{X}, \ | \, f(\mathbf{x}) \, | \leq M_f$, and the sign function, sgn, is defined as

$$sgn(x) \triangleq \begin{cases} 1, & if \quad x \geq 0 \ . \\ -1, & if \quad x < 0 \ . \end{cases} \tag{9.25}$$

9.4 Adaptive Law for Parameter Vector $\overline{\mathbf{Y}}$

Substituting eq.(9.22) in eq.(9.3) and using eq.(9.19) gives

$$\begin{cases} \dot{\epsilon}_1 &= \epsilon_2 \ . \\ \vdots & \\ \dot{\epsilon}_{N-1} &= \epsilon_N \ . \\ \dot{\epsilon}_N &= -\alpha\epsilon_N - \sum_{k=1}^{N-1} \alpha_k \epsilon_k - \mathbf{\Theta}^T(\mathbf{x},0)\mathbf{\Delta}_{\overline{\mathbf{Y}}} - \Delta_g u_c - r(\mathbf{x},0,\mathbf{\Theta},\overline{\mathbf{Y}}^*) \\ & \quad -S_u g u_s \ . \end{cases} \tag{9.26}$$

where Δ_g is the estimation error of the function $g(\mathbf{x})$.

$$\Delta_g \hat{=} \hat{g} - g \ . \tag{9.27}$$

and

$$\begin{aligned} r(\mathbf{x},0,\mathbf{\Theta},\overline{\mathbf{Y}}) &\hat{=} r(\mathbf{x},u,\mathbf{\Theta},\overline{\mathbf{Y}})\,|_{u\equiv 0} \\ &= \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} - \alpha x_N - f(x) \ . \end{aligned} \tag{9.28}$$

Let

$$\alpha_N \hat{=} \alpha \ . \tag{9.29}$$

then equation (9.26) can be written in matrix form as.

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}[\, -\mathbf{\Theta}^T(\mathbf{x},0)\mathbf{\Delta}_{\overline{\mathbf{Y}}} - \Delta_g u_c - r(\mathbf{x},0,\mathbf{\Theta},\overline{\mathbf{Y}}^*) - S_u g u_s \,] \ . \tag{9.30}$$

where

$$\mathbf{e} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_{N-1} \\ \epsilon_N \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -\alpha_1 & -\alpha_2 & -\alpha_3 & \dots & -\alpha_N \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

The parameters, $\{\alpha_1, \ldots, \alpha_N\}$, are chosen in such a way that $\mathbf{A}$ is a stable matrix, i.e., the roots of the polynomial,

$$\det[s\mathbf{I} - \mathbf{A}] = s^N + \alpha_N s^{N-1} + \cdots + \alpha_1 \ , \tag{9.31}$$

are all located in the open left half of the complex plane.

Following the Lyapunov synthesis approach, the adaptive laws for the parameter vector $\overline{\mathbf{Y}}$, and for the estimate, $\hat{g}$, of the function, $g(\mathbf{x})$, can be derived, and stability of the control system guaranteed. The estimate, $\hat{g}$, of the unknown function, $g(\mathbf{x})$, can be obtained either with a self-tuning scheme, or with a static FLS.

In the following, we first present the adaptive laws for $\overline{\mathbf{Y}}$ and $\hat{g}$, and then analyse system stability properties in the form of theorems.

This adaptive law for $\overline{\mathbf{Y}}$ is

$$\dot{\overline{\mathbf{Y}}} = -\mathbf{H}[\mathbf{\Theta}(\mathbf{x}, u)h\xi - \mathbf{\Theta}(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] - S\beta\mathbf{H}\overline{\mathbf{Y}} \ . \tag{9.32}$$

where $\mathbf{H}$ is a constant positive definite symmetric matrix, and h is a positive constant that weights the identification error. $\mathbf{P}$ is a positive definite symmetric matrix that satisfies the Lyapunov equation,

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}^T\mathbf{A} = -\mathbf{Q} \ . \tag{9.33}$$

where $\mathbf{Q}$ is some given positive definite symmetric matrix.

The last term on the right hand side of eq.(9.32) represents a projection algorithm modification [42, 57, 133], to avoid circumstances under which the parameter vector, $\overline{\mathbf{Y}}$, becomes too large or even drifts to infinity [133]. The *on* or *off* status of this term

is controlled by the switch S, which is defined as

$$S = \begin{cases} 0\ , & if\ \| \overline{\mathbf{Y}} \| < M_{\overline{\mathbf{Y}}}, \\ & or\ \| \overline{\mathbf{Y}} \| = M_{\overline{\mathbf{Y}}}\ and\ \overline{\mathbf{Y}}^T \mathbf{H}[\mathbf{\Theta}(\mathbf{x}, u)h\xi - \mathbf{\Theta}(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] \geq 0 : \\ & or\ \| \overline{\mathbf{Y}} \| > M_{\overline{\mathbf{Y}}}\ and\ \overline{\mathbf{Y}}^T \mathbf{H}[\mathbf{\Theta}(\mathbf{x}, u)h\xi - \mathbf{\Theta}(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] > 0 : \\ 1\ , & otherwise: \end{cases} \tag{9.34}$$

β is a positive design parameter which satisfies

$$\begin{cases} \beta \geq \frac{\overline{\mathbf{Y}}^T \mathbf{H}[\mathbf{\Theta}(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e}) - \mathbf{\Theta}(\mathbf{x}, u)h\xi]}{\overline{\mathbf{Y}}^T \mathbf{H} \overline{\mathbf{Y}}}\ , & if\ \| \overline{\mathbf{Y}} \| = M_{\overline{\mathbf{Y}}}\ and \\ & \overline{\mathbf{Y}}^T \mathbf{H}[\mathbf{\Theta}(\mathbf{x}, u)h\xi - \mathbf{\Theta}(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] < 0 : \\ \beta > \frac{\overline{\mathbf{Y}}^T \mathbf{H}[\mathbf{\Theta}(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e}) - \mathbf{\Theta}(\mathbf{x}, u)h\xi]}{\overline{\mathbf{Y}}^T \mathbf{H} \overline{\mathbf{Y}}}\ , & if\ \| \overline{\mathbf{Y}} \| > M_{\overline{\mathbf{Y}}}\ and \\ & \overline{\mathbf{Y}}^T \mathbf{H}[\mathbf{\Theta}(\mathbf{x}, u)h\xi - \mathbf{\Theta}(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] \leq 0 : \end{cases} \tag{9.35}$$

9.5 Adaptive Laws for $\hat{g}$

Two approaches are presented in this work to estimate $g(\mathbf{x})$. One is based on a traditional self-tuning scheme, the other is a FLS approach. If there is no expert knowledge about $g(\mathbf{x})$ available at all, the former approach may be more appropriate, since it is more straight forward and simpler than the other. If, however, there is expert knowledge of $g(\mathbf{x})$ available, the FLS approach may be more advantageous, since it can systematically incorporate human linguistic information, and result in better system performance.

9.5.1 Self-Tuning Scheme for $\hat{g}$

The adaptive law for $\hat{g}$ is

$$\dot{\hat{g}} = \gamma(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c + S_g \beta_g \gamma \hat{g}\ . \tag{9.36}$$

where γ is a positive gain. S_g is a switch defined as

$$S_g \triangleq \begin{cases} 0 . & if \quad \varepsilon_g < |\dot{g}| < M_g . \\ & or \quad |\dot{g}| = \varepsilon_g \quad and \quad \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \geq 0 . \\ & or \quad |\dot{g}| = M_g \quad and \quad \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \leq 0 . \\ & or \quad |\dot{g}| > M_g \quad and \quad \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c < 0 . \\ -1 . & if \quad |\dot{g}| = M_g \quad and \quad \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c > 0 . \\ & or \quad |\dot{g}| > M_g \quad and \quad \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \geq 0 . \\ 1 . & if \quad |\dot{g}| = \varepsilon_g \quad and \quad \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c < 0 . \end{cases} \tag{9.37}$$

and β_g is a positive design parameter which satisfies

$$\begin{cases} \beta_g \geq \pm\frac{\dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c}{\dot{g}^2} . & if \quad |\dot{g}| = M_g \quad and \quad \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c > 0 \ (for\ positive\ sign) . \\ & or \quad |\dot{g}| = \varepsilon_g \quad and \quad \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c < 0 \ (for\ negative\ sign) . \\ \beta_g > \frac{\dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c}{\dot{g}^2} . & if \quad |\dot{g}| > M_g \quad and \quad \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \geq 0 . \end{cases} \tag{9.38}$$

It is possible that $|\dot{g}|$ exceeds its lower bound, ε_g, during the training process in the following way: as the training is implemented computationally as

$$\dot{g}(t + dt) = \dot{g}(t) + \ddot{g}(t)dt . \tag{9.39}$$

(where $\ddot{g}(t)$ is given by the training law, eq.(9.36)), then if $|\dot{g}(t)|$ is very close to its lower bound, ε_g, but still within this bound, the resulting $|\dot{g}(t + dt)|$ can possibly exceed this prespecified lower bound, ε_g.

In this special situation, we can change the value of the gain, γ, in eq.(9.36) and obtain a new value for $\ddot{g}(t)$, which would result in a new value of $\dot{g}(t + dt)$ (such that $|\dot{g}(t + dt)| \geq \varepsilon_g$). Another alternative could simply be to let $\dot{g}(t + dt) \triangleq sgn[\dot{g}(t)]\varepsilon_g$ to skip this undesired situation, and carry on the adaptive calculation with new samples.

9.5.2 FLS Approach for $\hat{g}$

Since the function $g(\mathbf{x})$ is of a static nature, a static FLS is used as its estimator. Consider the FLS defined in eqs.(4.43–4.45). Let

$$\hat{g} = \mathbf{\Psi}^T(\mathbf{x})\overline{\mathbf{G}} \ . \tag{9.40}$$

where

$$\begin{cases} \mathbf{\Psi}(\mathbf{x}) & \triangleq & \{\psi_1(\mathbf{x}), \ \ldots \ , \psi_J(\mathbf{x})\}^T \ . \\ \overline{\mathbf{G}} & \triangleq & \{\overline{g}_1, \ \ldots \ , \overline{g}_J\}^T \ . \end{cases} \tag{9.41}$$

and the membership functions are defined as

$$\psi_j(\mathbf{x}) \triangleq \frac{\prod_{p=1}^{P} \exp[-(\frac{x_p - \overline{x}^j_{jp}}{\sigma^j_{jp}})^2]}{\sum_{j=1}^{J} \prod_{p=1}^{P} \exp[-(\frac{x_p - \overline{x}^j_{jp}}{\sigma^j_{jp}})^2]} \ . \tag{9.42}$$

Equations (9.40–9.42) represent a static FLS characterized by a singleton fuzzifier, centroid defuzzifier, sup-product compositional rule of inference, algebraic product operation for fuzzy implication and t-norm, and Gaussian type membership functions for primary fuzzy sets.

In eq.(9.40), $\mathbf{\Psi}(\mathbf{x})$ is designed off-line and $\overline{\mathbf{G}}$ is the parameter vector adjusted on-line. Before presenting the adaptive law, we impose an assumption on $\hat{g}$ and $\overline{\mathbf{G}}$.

Assumption 9.2 *$\forall \ x \in \mathcal{X}$, if $\| \overline{\mathbf{G}} \| \geq M_{\overline{\mathbf{G}}}$, then $| \hat{g} | \geq \varepsilon_g$, where $M_{\overline{\mathbf{G}}}$ is some large positive constant.*

The adaptive law for parameter vector, $\overline{\mathbf{G}}$, is

$$\dot{\overline{\mathbf{G}}} = \gamma \mathbf{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c - S_g \beta_g \gamma \overline{\mathbf{G}} \ . \tag{9.43}$$

where γ is a positive gain, S_g is a switch to control the *on* or *off* status of the projection modification term, and β_g is a positive design parameter.

$$\{S_g, \ \beta_g\}^T \triangleq \begin{cases} \{S_g^U, \ \beta_g^U\}^T \ . & \text{if} \quad | \hat{g} | > \varepsilon_g \ . \\ \{-S_g^L, \ \beta_g^L\}^T \ . & \text{if} \quad | \hat{g} | = \varepsilon_g \quad \text{and} \quad \| \overline{\mathbf{G}} \| < M_{\overline{\mathbf{G}}} \ . \end{cases} \tag{9.44}$$

where

$$S_g^U \triangleq \begin{cases} 0\ . \ \ if\ \ \| \overline{\mathbf{G}} \| < M_{\overline{\mathbf{G}}}, \\ \qquad or\ \ \| \overline{\mathbf{G}} \| = M_{\overline{\mathbf{G}}}\ \ and\ \ \overline{\mathbf{G}}^T \boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \leq 0\ . \\ \qquad or\ \ \| \overline{\mathbf{G}} \| > M_{\overline{\mathbf{G}}}\ \ and\ \ \overline{\mathbf{G}}^T \boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c < 0\ . \\ \\ 1\ . \ \ otherwise. \end{cases} \tag{9.45}$$

$$S_g^L \triangleq \begin{cases} 0\ . \ \ if\ \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \geq 0\ . \\ \\ 1\ . \ \ if\ \dot{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c < 0\ . \end{cases} \tag{9.46}$$

$$\begin{cases} \beta_g^U \geq \dfrac{\overline{\mathbf{G}}^T \boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c}{\overline{\mathbf{G}}^T \overline{\mathbf{G}}}\ . \ \ if\ \ \| \overline{\mathbf{G}} \| = M_{\overline{\mathbf{G}}}\ \ and\ \ \overline{\mathbf{G}}^T \boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c > 0\ : \\ \\ \beta_g^U > \dfrac{\overline{\mathbf{G}}^T \boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c}{\overline{\mathbf{G}}^T \overline{\mathbf{G}}}\ . \ \ if\ \ \| \overline{\mathbf{G}} \| > M_{\overline{\mathbf{G}}}\ \ and\ \ \overline{\mathbf{G}}^T \boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \geq 0\ . \end{cases} \tag{9.47}$$

$$\beta_g^L \geq -\frac{\dot{g}(\boldsymbol{\Psi}^T\boldsymbol{\Psi})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c}{\dot{g}^2}\ . \tag{9.48}$$

$M_{\overline{\mathbf{G}}}$ is a positive design constant that bounds $\| \overline{\mathbf{G}} \|$, and satisfies

$$M_{\overline{\mathbf{G}}} \leq \frac{1}{\sqrt{J}} M_g\ . \tag{9.49}$$

where J is the size of the vectors, $\boldsymbol{\Psi}$ and $\overline{\mathbf{G}}$, as in eq.(9.41).

As above, it is possible that $| \dot{g} | < \varepsilon_g$ and $\| \overline{\mathbf{G}} \| < M_{\overline{\mathbf{G}}}$, during training, where again as the training is implemented computationally as

$$\hat{g}(t + dt) = \hat{g}(t) + \dot{\hat{g}}(t)dt = \hat{g}(t) + \boldsymbol{\Psi}^T \dot{\overline{\mathbf{G}}}(t)dt\ . \tag{9.50}$$

(where $\dot{\overline{\mathbf{G}}}(t)$ is given by the training law, eq.(9.43)), then if $| \dot{g}(t) |$ is very close to its lower bound, ε_g, but still within this bound, the resulting $| \dot{g}(t + dt) |$ can possibly exceed this prespecified lower bound.

In this situation, we can change the value of the gain, γ, and obtain a new value of $\dot{\overline{\mathbf{G}}}(t)$, which would result in a new value of $\hat{g}(t+dt)$ (such that $|\hat{g}(t+dt)| \geq \varepsilon_g$). Another choice could simply be to let $\hat{g}(t+dt) \triangleq sgn[\hat{g}(t)]\varepsilon_g$ to skip this situation, and carry on the adaptive calculation with new samples.

9.6 Stability Properties of DFLS Control Algorithm

The stability properties of the DFLS based adaptive control system are summarized in the following theorem.

Theorem 9.1 *Consider an unknown nonlinear dynamic system described by eq.(9.1), which is subject to the control law, eq.(9.20). If the DFLS of eq.(9.4) is used to identify the state, x_N, with adaptive law, eq.(9.32), for tuning its parameter vector, $\overline{\mathbf{Y}}$, and furthermore the unknown function, $g(x)$, is estimated either by a self-tuning scheme with adaptive law, eq.(9.36), or by a FLS, with adaptive law, eq.(9.43), then the closed loop system has the following stability properties.*

9.1.1 $\| \overline{\mathbf{Y}} \| \leq M_{\overline{\mathbf{Y}}}$;

9.1.2 $\varepsilon_g \leq |\hat{g}| \leq M_g$;

9.1.3 $\| \mathbf{e} \| \leq M_{\mathbf{e}}$, $\| \mathbf{x} \| \leq M_{\mathbf{e}} + \| \mathbf{q} \|$, *and* $|\xi| \leq M_\xi$, *where* $\mathbf{q} \triangleq \{q_1, \ldots, q_N\}^T$ *and* M_ξ *is a positive constant;*

9.1.4

$$|u(t)| \leq \frac{1}{\varepsilon_g} \cdot \{ (2 + \frac{M_g + |\hat{g}(t)|}{\varepsilon_g}) \cdot (\sqrt{l} M_{\overline{\mathbf{Y}}} + a \cdot |x_N(t)|) + (1 + \frac{M_g + |\hat{g}(t)|}{\varepsilon_g}) \cdot (|\dot{q}_N(t)| + \| \mathbf{a} \| M_{\mathbf{e}}) + M_f \} .$$

where $\mathbf{a} \triangleq \{\alpha_1, \ldots, \alpha_N\}^T$. *I is the size of the vector* $\boldsymbol{\Theta}$, *and* M_f *is a positive constant that bounds* $| f(\mathbf{x}) |$:

9.1.5

$$\begin{aligned} \int_0^t \| \mathbf{e}(\tau) \|^2 \, d\tau &\leq b + \eta \int_0^t [\, r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2 \,] d\tau \, . \\ \int_0^t \xi^2(\tau) d\tau &\leq b + \eta \int_0^t [\, r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2 \,] d\tau \, . \end{aligned}$$

where b *and* η *are positive constants;*

9.1.6 *if* $r(t) \in \mathcal{L}_2[0, \infty)$ *and* $\Delta_g \in \mathcal{L}_2[0, \infty)$, *i.e.*, $(\int_0^\infty \| r(\tau) \|^2 \, d\tau)^{\frac{1}{2}} < \infty$ *and* $(\int_0^\infty \Delta_g^2 d\tau)^{\frac{1}{2}} < \infty$, *then,*

$$\begin{aligned} \lim_{t \to \infty} \| \mathbf{e}(t) \| &= 0 \, . \\ \lim_{t \to \infty} | \xi(t) | &= 0 \, . \end{aligned}$$

the proofs will be addressed presently, but we first consider a particular case.

9.7 A Special Situation : $g(\mathbf{x})$ Known

Consider the dynamic system described by eq.(9.1),

$$x^{(N)} = f(\mathbf{x}) + g(\mathbf{x}) u \, . \tag{9.51}$$

It is possible that the function, g, is known to us. In this situation, the indirect adaptive control algorithm developed previously is still applicable, provided that the identifier to estimate the function $g(x)$ is eliminated, and

$$\begin{cases} \hat{g} \equiv g \, . \\ \Delta_g \equiv 0 \, . \end{cases} \tag{9.52}$$

Theorem 9.1 of course still holds, but Theorem 9.1.2 is unnecessary here. The results for this special situation are summarized in the following corollaries. The control terms, u_c and u_s, are given as follows:

$$u_c \triangleq \frac{1}{g}[\, \dot{q}_N + \alpha q_N - \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} - \sum_{k=1}^{N-1} \alpha_k \epsilon_k \,] \,. \tag{9.53}$$

$$u_s \triangleq sgn(\mathbf{B}^T\mathbf{P}\mathbf{e}) \cdot \frac{1}{g} \cdot [\, \alpha \mid x_N \mid + \mid \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} \mid + M_f \,] \,. \tag{9.54}$$

Corollary 9.1 *Consider a nonlinear system described by eq.(9.51), which is the same as eq.(9.1) except that g is a known function here, and the control terms, u_c and u_s, in the control law, eq.(9.20), are given in eqs.(9.53-9.54). Then, the closed loop system has the following stability properties.*

- $\| \overline{\mathbf{Y}} \| \leq M_{\overline{\mathbf{Y}}}$;
- $\| \mathbf{e} \| \leq M_{\mathbf{e}}$, $\| \mathbf{x} \| \leq M_{\mathbf{e}} + \| \mathbf{q} \|$, *and* $\mid \xi \mid \leq M_\xi$;
- $$\begin{cases} \int_0^t \| \mathbf{e}(\tau) \|^2 d\tau & \leq b + \eta \int_0^t [\, r^2(\mathbf{x},0,\mathbf{\Theta},\overline{\mathbf{Y}}) + r^2(\mathbf{x},u,\mathbf{\Theta},\overline{\mathbf{Y}}) \,] d\tau \,, \\ \int_0^t \xi^2(\tau) d\tau & \leq b + \eta \int_0^t [\, r^2(\mathbf{x},0,\mathbf{\Theta},\overline{\mathbf{Y}}) + r^2(\mathbf{x},u,\mathbf{\Theta},\overline{\mathbf{Y}}) \,] d\tau \,, \end{cases}$$
  where b and η are positive constants;
- *if $r(t) \in \mathcal{L}_2[0,\infty)$, then, $\lim_{t\to\infty} \| \mathbf{e}(t) \| = 0$, $\lim_{t\to\infty} \mid \xi(t) \mid = 0$.*

Corollary 9.2

$$\mid u(t) \mid \leq \frac{1}{\mid g \mid} \cdot [\, 2 \cdot (\sqrt{l} M_{\overline{\mathbf{Y}}} + \alpha \mid x_N(t) \mid) + \mid \dot{q}_N(t) \mid + \| \mathbf{a} \| M_{\mathbf{e}} + M_f \,] \,.$$

9.8 Proof of Theorem and Corollaries

9.8.1 Proof of Theorem 9.1.1

Consider a Lyapunov function candidate,

$$V_{\overline{\mathbf{Y}}} = \frac{1}{2}\overline{\mathbf{Y}}^T\overline{\mathbf{Y}} \,. \tag{9.55}$$

Differentiating both sides of eq.(9.55) yields

$$\dot{V}_{\overline{\mathbf{Y}}} = \overline{\mathbf{Y}}^T \dot{\overline{\mathbf{Y}}} \ . \tag{9.56}$$

Using eq.(9.32) in eq.(9.56) yields

$$\dot{V}_{\overline{\mathbf{Y}}} = -\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, u)h\xi - \Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] - S\beta\overline{\mathbf{Y}}^T\mathbf{H}\overline{\mathbf{Y}} \ . \tag{9.57}$$

We have mentioned that $M_{\overline{\mathbf{Y}}} > 0$ is a design constant which bounds the norm of $\overline{\mathbf{Y}}$. Here, we call $M_{\overline{\mathbf{Y}}}$ the *nominal bound* of $\| \overline{\mathbf{Y}} \|$. Now, we discuss different situations.

(i) $\| \overline{\mathbf{Y}} \| < M_{\overline{\mathbf{Y}}}$:

This is the desired situation, and the norm of the parameter vector, $\overline{\mathbf{Y}}$, is of no concern to us.

(ii) $\| \overline{\mathbf{Y}} \| = M_{\overline{\mathbf{Y}}}$:

Consider eq.(9.34).

- for $\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, u)h\xi - \Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] \geq 0$, $S = 0$,

$$\dot{V}_{\overline{\mathbf{Y}}} = -\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, u)h\xi - \Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] \leq 0 \ . \tag{9.58}$$

- for $\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, u)h\xi - \Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] < 0$, $S = 1$, and

$$\dot{V}_{\overline{\mathbf{Y}}} = -\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, u)h\xi - \Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] - \beta\overline{\mathbf{Y}}^T\mathbf{H}\overline{\mathbf{Y}} \ , \tag{9.59}$$

$$\beta \geq \frac{\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e}) - \Theta(\mathbf{x}, u)h\xi]}{\overline{\mathbf{Y}}^T\mathbf{H}\overline{\mathbf{Y}}} \ . \tag{9.60}$$

or,

$$\dot{V}_{\overline{\mathbf{Y}}} \leq 0 \ . \tag{9.61}$$

Equations (9.58) and (9.61) indicate that $\| \overline{\mathbf{Y}} \|$ tends to increase, and the boundedness of $\| \overline{\mathbf{Y}} \|$ can be concluded.

(iii) $\| \overline{\mathbf{Y}} \| > M_{\overline{\mathbf{Y}}}$:

In this case, $\| \overline{\mathbf{Y}} \|$ exceeds its *nominal bound*, $M_{\overline{\mathbf{Y}}}$, which may occur during the training process, and as the training is implemented computationally as

$$\overline{\mathbf{Y}}(t + dt) = \overline{\mathbf{Y}}(t) + \dot{\overline{\mathbf{Y}}}(t)dt \ . \tag{9.62}$$

(where $\dot{\overline{\mathbf{Y}}}(t)$ is given by the training law eq.(9.32)), then in the situation in which $\| \overline{\mathbf{Y}}(t) \|$ is very close to the nominal bound, $M_{\overline{\mathbf{Y}}}$, but still within this bound, the resulting $\| \overline{\mathbf{Y}}(t + dt) \|$ can exceed the prespecified nominal bound by the norm of the training increment, $\dot{\overline{\mathbf{Y}}}(t)dt$, at most. Considering eq.(9.34), then,

– for $\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, u)h\xi - \Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] > 0$, $S = 0$,

$$\dot{V}_{\overline{\mathbf{Y}}} = -\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, u)h\xi - \Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] < 0 \ . \tag{9.63}$$

– for $\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, u)h\xi - \Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] \leq 0$, $S = 1$, and

$$\dot{V}_{\overline{\mathbf{Y}}} = -\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, u)h\xi - \Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] - \beta\overline{\mathbf{Y}}^T\mathbf{H}\overline{\mathbf{Y}} \ , \tag{9.64}$$

$$\beta > \frac{\overline{\mathbf{Y}}^T \mathbf{H}[\Theta(\mathbf{x}, 0)(\mathbf{B}^T\mathbf{P}\mathbf{e}) - \Theta(\mathbf{x}, u)h\xi]}{\overline{\mathbf{Y}}^T\mathbf{H}\overline{\mathbf{Y}}} \ . \tag{9.65}$$

or,

$$\dot{V}_{\overline{\mathbf{Y}}} < 0 \ . \tag{9.66}$$

Equations (9.63) and (9.66) indicate that as soon as $\| \overline{\mathbf{Y}} \|$ exceeds its nominal bound, $M_{\overline{\mathbf{Y}}}$, it immediately returns within this bound. Therefore, this excess is *small* and *temporary*. Formally, denote

$$\delta M_{\overline{\mathbf{Y}}} \triangleq \sup_{\{\mathbf{x}, u\} \in \Xi} \| \dot{\overline{\mathbf{Y}}}(t)dt \| \ . \tag{9.67}$$

and let

$$M'_{\overline{\mathbf{Y}}} \triangleq M_{\overline{\mathbf{Y}}} + \delta M_{\overline{\mathbf{Y}}} \ . \tag{9.68}$$

then we have

$$\| \overline{\mathbf{Y}}(t+dt) \| \leq M'_{\overline{\mathbf{Y}}} . \tag{9.69}$$

which, again, indicates the boundedness of $\overline{\mathbf{Y}}$.

For simplicity but without loss of generality, we still use $M_{\overline{\mathbf{Y}}}$ instead of $M'_{\overline{\mathbf{Y}}}$ in this work, which is equivalent to implicitly specifying that $M_{\overline{\mathbf{Y}}} \triangleq \max\{M_{\overline{\mathbf{Y}}}, M'_{\overline{\mathbf{Y}}}\}$.

This concludes the proof of Theorem 9.1.1.

9.8.2 Proof of Theorem 9.1.2 – Estimation of $g(\mathbf{x})$ with Self-tuning Scheme

Consider a Lyapunov function candidate,

$$V_g = \frac{1}{2}\tilde{g}^2 . \tag{9.70}$$

Differentiating both sides of eq.(9.70) yields

$$\dot{V}_g = \tilde{g}\dot{\tilde{g}} . \tag{9.71}$$

Using eq.(9.36) in (9.71) yields

$$\dot{V}_g = \gamma\tilde{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c + S_g\gamma\beta_g\tilde{g}^2 . \tag{9.72}$$

(i) $|\hat{g}| = \varepsilon_g$:

Consider eq.(9.37).

– for $\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \geq 0$, $S_g = 0$.

$$\dot{V}_g = \gamma\tilde{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \geq 0 . \tag{9.73}$$

– for $\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c < 0$, $S_g = 1$, and

$$\dot{V}_g = \gamma\tilde{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c + \gamma\beta_g\tilde{g}^2 , \tag{9.74}$$

$$\beta_g \geq -\frac{\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c}{\hat{g}^2} . \tag{9.75}$$

or,

$$\dot{V}_g \geq 0 \ . \tag{9.76}$$

Equations (9.73) and (9.76) indicate that $| \hat{g} |$ tends to increase, and therefore, the lower boundedness of $| \hat{g} |$ can be concluded.

(ii) $\varepsilon_g < | \hat{g} | < M_g$:

This is the desired situation, and $| \hat{g} |$ is of no concern to us.

(iii) $| \hat{g} | = M_g$:

Consider eq.(9.37),

- for $\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \leq 0$, $S_g = 0$,

$$\dot{V}_g = -\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \leq 0 \ . \tag{9.77}$$

- for $\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c > 0$, $S_g = -1$, and

$$\dot{V}_g = -\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c - \beta_g\dot{\hat{g}}^2 \ , \tag{9.78}$$

$$\beta_g \geq \frac{\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c}{\dot{\hat{g}}^2} \ . \tag{9.79}$$

or,

$$\dot{V}_g \leq 0 \ . \tag{9.80}$$

Both eqs.(9.77) and (9.80) indicate that $| \hat{g} |$ tends to decrease, and therefore, its upper boundedness can be concluded.

(iv) $| \hat{g} | > M_g$:

In this case, $| \hat{g} |$ exceeds its upper bound, M_g, which may occur during the training process, and as the training is implemented computationally as

$$\hat{g}(t + dt) = \hat{g}(t) + \dot{\hat{g}}(t)dt \ . \tag{9.81}$$

(where $\dot{\hat{g}}(t)$ is given by the training law eq.(9.36)), then in the situation in which $|\hat{g}(t)|$ is very close to its upper bound, $M_{\hat{g}}$, but still within this bound, the resulting $|\hat{g}(t+dt)|$ can exceed the prespecified bound, $M_{\hat{g}}$, by $|\dot{\hat{g}}(t)dt|$, at most. Considering eq.(9.37), then

- for $\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c < 0$, $S_{\hat{g}} = 0$,

$$\dot{V}_{\hat{g}} = \gamma|\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c| < 0 \ . \tag{9.82}$$

- for $\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \geq 0$, it has $S_{\hat{g}} = -1$, and

$$\dot{V}_{\hat{g}} = \gamma\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c - \gamma\beta_{\hat{g}}\dot{\hat{g}}^2 \ , \tag{9.83}$$

$$\beta_{\hat{g}} > \frac{\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c}{\dot{\hat{g}}^2} \ . \tag{9.84}$$

or,

$$\dot{V}_{\hat{g}} < 0 \ . \tag{9.85}$$

Equations (9.82) and (9.85) indicate that as soon as $|\hat{g}|$ exceeds its upper bound, $M_{\hat{g}}$, it immediately returns within this bound. Therefore, this excess is *small* and *temporary*. Formally, denote

$$\delta M_{\hat{g}} \triangleq \sup_{\mathbf{x}\in X} |\dot{\hat{g}}(t)dt| \ , \tag{9.86}$$

and let

$$M'_{\hat{g}} \triangleq M_{\hat{g}} + \delta M_{\hat{g}} \ , \tag{9.87}$$

then we have

$$|\hat{g}(t+dt)| \leq M'_{\hat{g}} \ , \tag{9.88}$$

which, again, indicates the boundedness of $|\hat{g}|$.

For simplicity but without loss of generality, we still use $M_{\hat{g}}$ instead of $M'_{\hat{g}}$ in this work, which is equivalent to implicitly define that $M_{\hat{g}} \triangleq \max\{M_{\hat{g}}, M'_{\hat{g}}\}$.

This concludes the proof of Theorem 9.1.2 for the situation that the unknown function, $g(\mathbf{x})$, is estimated with a self-tuning scheme.

9.8.3 Proof of Theorem 9.1.2 – Estimation of $g(\mathbf{x})$ with FLS

(i) $|\dot{g}| > \varepsilon_g$:

Consider a Lyapunov function candidate,

$$V_{\overline{\mathbf{G}}} = \frac{1}{2}\overline{\mathbf{G}}^T\overline{\mathbf{G}} . \tag{9.89}$$

Differentiating both sides of eq.(9.89) yields,

$$\dot{V}_{\overline{\mathbf{G}}} = \overline{\mathbf{G}}^T\dot{\overline{\mathbf{G}}} . \tag{9.90}$$

Using eq.(9.43) in eq.(9.90), and considering eq.(9.44) gives

$$\dot{V}_{\overline{\mathbf{G}}} = -\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c - S_g^U\beta_g^U\overline{\mathbf{G}}^T\overline{\mathbf{G}} . \tag{9.91}$$

Let $0 < M_{\overline{\mathbf{G}}} < \infty$ be the nominal bound of the norm of $\overline{\mathbf{G}}$. We now consider different situations.

- $\| \overline{\mathbf{G}} \| < M_{\overline{\mathbf{G}}}$:

  This is the desired situation, and the norm of the parameter vector, $\overline{\mathbf{G}}$, is of no concern to us.

- $\| \overline{\mathbf{G}} \| = M_{\overline{\mathbf{G}}}$:

  Considering eq.(9.45),

  * for $\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \leq 0$, $S_g^U = 0$.

$$\dot{V}_{\overline{\mathbf{G}}} = -\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c \leq 0 . \tag{9.92}$$

* for $\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{Pe})u_c > 0$, $S_g^U = 1$, and

$$\dot{V}_{\overline{\mathbf{G}}} = \gamma\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{Pe})u_c - \beta_g^U\gamma\overline{\mathbf{G}}^T\overline{\mathbf{G}} \,. \tag{9.93}$$

$$\beta_g^U \geq \frac{\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{Pe})u_c}{\overline{\mathbf{G}}^T\overline{\mathbf{G}}} \,. \tag{9.94}$$

or,

$$\dot{V}_{\overline{\mathbf{G}}} \leq 0 \,. \tag{9.95}$$

Equations (9.92) and (9.95) indicate that $\| \overline{\mathbf{G}} \|$ tends to decrease, and therefore, the boundedness of $\| \overline{\mathbf{G}} \|$ can be concluded.

- $\| \overline{\mathbf{G}} \| > M_{\overline{\mathbf{G}}}$:

In this case, $\| \overline{\mathbf{G}} \|$ exceeds its nominal bound, $M_{\overline{\mathbf{G}}}$, which may occur during the training process, and as the training is implemented computationally as

$$\overline{\mathbf{G}}(t+dt) = \overline{\mathbf{G}}(t) - \dot{\overline{\mathbf{G}}}(t)dt \,. \tag{9.96}$$

(where $\dot{\overline{\mathbf{G}}}(t)$ is given by the training law eq.(9.43)), then in the situation in which $\| \overline{\mathbf{G}}(t) \|$ is very close to the nominal bound, $M_{\overline{\mathbf{G}}}$, but still within this bound, the resulting $\| \overline{\mathbf{G}}(t+dt) \|$ can exceed the prespecified nominal bound by the norm of the training increment, $\dot{\overline{\mathbf{G}}}(t)dt$, at most. Considering eq.(9.45), then,

* for $\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{Pe})u_c < 0$, $S_g^U = 0$.

$$\dot{V}_{\overline{\mathbf{G}}} = \gamma\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{Pe})u_c < 0 \,. \tag{9.97}$$

* for $\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{Pe})u_c \geq 0$, $S_g^U = 1$, and

$$\dot{V}_{\overline{\mathbf{G}}} = \gamma\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{Pe})u_c - \beta_g^U\gamma\overline{\mathbf{G}}^T\overline{\mathbf{G}} \,. \tag{9.98}$$

$$\beta_g^U > \frac{\overline{\mathbf{G}}^T\boldsymbol{\Psi}(\mathbf{x})(\mathbf{B}^T\mathbf{Pe})u}{\overline{\mathbf{G}}^T\overline{\mathbf{G}}} \,. \tag{9.99}$$

or,

$$\dot{V}_{\overline{\mathbf{G}}} < 0 \,. \tag{9.100}$$

Equations (9.97) and (9.100) indicate that as soon as $\| \overline{\mathbf{G}} \|$ exceeds its nominal bound, $M_{\overline{\mathbf{G}}}$, it immediately returns within this bound. Therefore, this excess is *small* and *temporary*. Formally, denote

$$\delta M_{\overline{\mathbf{G}}} \triangleq \sup_{\mathbf{x} \in X} \| \dot{\overline{\mathbf{G}}}(t) dt \| \,. \tag{9.101}$$

and let

$$M'_{\overline{\mathbf{G}}} \triangleq M_{\overline{\mathbf{G}}} + \delta M_{\overline{\mathbf{G}}} \,. \tag{9.102}$$

then we have

$$\| \overline{\mathbf{G}}(t + dt) \| \leq M'_{\overline{\mathbf{G}}} \,. \tag{9.103}$$

which, again, indicates the boundedness of $\| \overline{\mathbf{G}} \|$.

For simplicity but without loss of generality, we still use $M_{\overline{\mathbf{G}}}$ instead of $M'_{\overline{\mathbf{G}}}$, which is equivalent to implicitly specifying that $M_{\overline{\mathbf{G}}} \triangleq \max\{M_{\overline{\mathbf{G}}}, M'_{\overline{\mathbf{G}}}\}$

Therefore, in situation (i), i.e., $| \hat{g} | > \varepsilon_g$, it is concluded that

$$\| \overline{\mathbf{G}} \| \leq M_{\overline{\mathbf{G}}} \,. \tag{9.104}$$

Considering eq.(9.40)

$$| \hat{g} | = | \mathbf{\Psi}^T \overline{\mathbf{G}} | \leq \| \mathbf{\Psi} \| \cdot \| \overline{\mathbf{G}} \| \,. \tag{9.105}$$

and it is known that of all elements of the vector, $\mathbf{\Psi}$, none is larger than one, i.e., $\psi_j \leq 1$, for $j = 1, \ldots, J$. Therefore,

$$\| \mathbf{\Psi} \| = \sqrt{\sum_{j=1}^{J} \psi_j^2} \leq \sqrt{J} \,. \tag{9.106}$$

Using eqs.(9.104) and (9.106) in eq.(9.105) yields

$$| \dot{g} | \leq \sqrt{J} M_{\overline{\mathbf{G}}} . \tag{9.107}$$

Considering eq.(9.49), i.e.,

$$M_{\overline{\mathbf{G}}} \leq \frac{1}{\sqrt{J}} M_g . \tag{9.108}$$

we have

$$| \dot{g} | \leq M_g . \tag{9.109}$$

(ii) $| \dot{g} | = \varepsilon_g$ and $\| \overline{\mathbf{G}} \| < M_{\overline{\mathbf{G}}}$:

In this situation, $\| \overline{\mathbf{G}} \|$ is of no concern to us. Consider a Lyapunov function candidate,

$$V_{\dot{g}} = \frac{1}{2} \dot{g}^2 . \tag{9.110}$$

Differentiating both sides of eq.(9.110) yields

$$\dot{V}_{\dot{g}} = \dot{g} \ddot{g} = \dot{g} \boldsymbol{\Psi}^T \dot{\overline{\mathbf{G}}} . \tag{9.111}$$

Considering eqs.(9.43-9.44), we have

$$\begin{aligned} \dot{V}_{\dot{g}} &= -\gamma \dot{g} \boldsymbol{\Psi}^T \boldsymbol{\Psi} (\mathbf{B}^T \mathbf{P} \mathbf{e}) u_c + S_g^L \beta_g^L \gamma \dot{g} (\boldsymbol{\Psi}^T \overline{\mathbf{G}}) \\ &= -\gamma \dot{g} \boldsymbol{\Psi}^T \boldsymbol{\Psi} (\mathbf{B}^T \mathbf{P} \mathbf{e}) u_c + S_g^L \beta_g^L \gamma \dot{g}^2 . \end{aligned} \tag{9.112}$$

Further, referring to eqs.(9.46) and (9.48),

- For $\dot{g} (\mathbf{B}^T \mathbf{P} \mathbf{e}) u_c \geq 0$, $S_g^L = 0$.

$$\dot{V}_{\dot{g}} = -\gamma \dot{g} \boldsymbol{\Psi}^T \boldsymbol{\Psi} (\mathbf{B}^T \mathbf{P} \mathbf{e}) u_c \geq 0 . \tag{9.113}$$

– For $\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c < 0$, $S_g^L = 1$, and

$$\dot{V}_{\hat{g}} = \gamma\hat{g}\boldsymbol{\Psi}^T\boldsymbol{\Psi}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c + \gamma\beta_g^L\dot{\hat{g}}^2 . \tag{9.114}$$

$$\beta_g^L \geq -\frac{(\boldsymbol{\Psi}^T\boldsymbol{\Psi})\hat{g}(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c}{\dot{\hat{g}}^2} . \tag{9.115}$$

or,

$$\dot{V}_{\hat{g}} \geq 0 . \tag{9.116}$$

Equations (9.113) and (9.116) indicate that $|\hat{g}|$ tends to increase, and therefore, the lower boundedness of $|\hat{g}|$ is concluded.

This concludes the proof of Theorem 9.1.2 for the situation of the unknown function, $g(\mathbf{x})$, being estimated with a FLS.

9.8.4 Proof of Theorem 9.1.3

Using eq.(9.28) in eq.(9.30) yields

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} - \mathbf{B}[-\boldsymbol{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} + \alpha x_N + f(\mathbf{x}) + g(\mathbf{x})u_c - \dot{g}u_c - S_u g u_s] . \tag{9.117}$$

Consider the Lyapunov function candidate,

$$V = \frac{1}{2}\mathbf{e}^T\mathbf{P}\mathbf{e} . \tag{9.118}$$

Differentiating both sides of eq.(9.118) yields

$$\dot{V} = \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{P}\mathbf{e} + \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} . \tag{9.119}$$

and use of eq.(9.117) in (9.119) yields

$$\begin{aligned}\dot{V} &= \frac{1}{2}\mathbf{e}^T(\mathbf{A}^T\mathbf{P} + \mathbf{P}^T\mathbf{A})\mathbf{e} + (\mathbf{B}^T\mathbf{P}\mathbf{e})[\alpha x_N + f(\mathbf{x}) + g(\mathbf{x})u_c - \boldsymbol{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} - \dot{g}u_c] \\ &\quad - S_u g(\mathbf{B}^T\mathbf{P}\mathbf{e})u_s \\ &= -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + (\mathbf{B}^T\mathbf{P}\mathbf{e})[\alpha x_N + f(\mathbf{x}) + g(\mathbf{x})u_c - \boldsymbol{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} - \dot{g}u_c] \\ &\quad - S_u g(\mathbf{B}^T\mathbf{P}\mathbf{e})u_s .\end{aligned} \tag{9.120}$$

Then

$$\dot{V}_e \leq -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + |\mathbf{B}^T\mathbf{P}\mathbf{e}| \cdot [\alpha |x_N| + |\mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}}| + M_f + (M_g + |\hat{g}|) \cdot |u_s|] - S_u g(\mathbf{B}^T\mathbf{P}\mathbf{e})u_s . \tag{9.121}$$

where M_f is the upper bound for $|f(\mathbf{x})|$. If $\|\mathbf{e}\| \geq M_e$, $S_u = 1$. Using eq.(9.24) with (9.121) yields

$$\dot{V}_e \leq -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + (1 - \frac{|g|}{\varepsilon_g}) \cdot |\mathbf{B}^T\mathbf{P}\mathbf{e}| \cdot [\alpha |x_N| + |\mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}}| + M_f + (M_g + |\hat{g}|) \cdot |u_s|] . \tag{9.122}$$

Since $|g| \geq \varepsilon_g$, then, for $\|\mathbf{e}\| \geq M_e$,

$$\dot{V}_e \leq -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} \leq 0 . \tag{9.123}$$

Therefore, when $\|\mathbf{e}\| \geq M_e$, $\dot{V}_e \leq 0$, which indicates the boundedness of $\|\mathbf{e}\|$, i.e.,

$$\|\mathbf{e}\| \leq M_e . \tag{9.124}$$

Further, since $\mathbf{e} = \mathbf{x} - \mathbf{q}$.

$$\begin{aligned} \|\mathbf{x}\| &\leq \|\mathbf{e}\| - \|\mathbf{q}\| \\ &\leq M_e - \|\mathbf{q}\| . \end{aligned} \tag{9.125}$$

Now, consider eq.(9.4),

$$\dot{y} = -\alpha y + \mathbf{\Theta}^T(\mathbf{x},u)\overline{\mathbf{Y}} . \tag{9.126}$$

The solution to eq.(9.126) is

$$y(t) = \epsilon^{-\alpha t}y(0) + \int_0^t \epsilon^{-\alpha(t-\tau)}\mathbf{\Theta}^T(\mathbf{x},u)\overline{\mathbf{Y}}d\tau . \tag{9.127}$$

or,

$$\begin{aligned} |y(t)| &\leq \epsilon^{-\alpha t}|y(0)| + \int_0^t \epsilon^{-\alpha(t-\tau)}|\mathbf{\Theta}^T(\mathbf{x},u)\overline{\mathbf{Y}}|d\tau . \\ &\leq \epsilon^{-\alpha t}|y(0)| + \epsilon^{-\alpha t}\int_0^t \epsilon^{\alpha\tau}\|\mathbf{\Theta}^T(\mathbf{x},u)\| \cdot \|\overline{\mathbf{Y}}\|d\tau . \end{aligned} \tag{9.128}$$

Consider

$$\begin{cases} \| \boldsymbol{\Theta}^T(\mathbf{x}, u) \| = \sqrt{\sum_{i=1}^{I} \theta_i^2(\mathbf{x}, u)} \leq \sqrt{I} . \\ \| \overline{\mathbf{Y}} \| \leq M_{\overline{\mathbf{Y}}} . \end{cases} \tag{9.129}$$

then

$$\begin{aligned} | y(t) | &\leq \epsilon^{-\alpha t} | y(0) | + \epsilon^{-\alpha t} \sqrt{I} M_{\overline{\mathbf{Y}}} \cdot \frac{1}{\alpha} \epsilon^{\alpha \tau} |_0^t \\ &= \epsilon^{-\alpha t} | y(0) | + \frac{1}{\alpha} \sqrt{I} M_{\overline{\mathbf{Y}}} (1 - \epsilon^{-\alpha t}) . \end{aligned} \tag{9.130}$$

For $t \geq 0$, $0 < \epsilon^{-\alpha t} \leq 1$, therefore,

$$| y(t) | \leq | y(0) | + \frac{1}{\alpha} \sqrt{I} M_{\overline{\mathbf{Y}}} . \tag{9.131}$$

Since $\xi = y - x$,

$$\begin{aligned} | \xi | &\leq | y(t) | + | x \\ &\leq | y(0) | + \frac{1}{\alpha} \sqrt{I} M_{\overline{\mathbf{Y}}} + M_{\mathbf{e}} + | \mathbf{q} | . \end{aligned} \tag{9.132}$$

Let

$$M_{\mathbf{q}} \hat{=} \sup_{t \geq 0} \{ | \mathbf{q}(t) | \} . \tag{9.133}$$

and

$$M_{\xi} \hat{=} | y(0) | + \frac{1}{\alpha} \sqrt{I} M_{\overline{\mathbf{Y}}} + M_{\mathbf{e}} + M_{\mathbf{q}} . \tag{9.134}$$

Then M_{ξ} is a positive constant that bounds $| \xi |$, i.e.,

$$| \xi | \leq M_{\xi} . \tag{9.135}$$

This completes the proof of Theorem 9.1.3.

9.8.5 Proof of Theorem 9.1.4

Equations (9.20) and (9.24) together give

$$u = u_c - S_u sgn(\mathbf{B}^T\mathbf{Pe}) \cdot sgn(g) \cdot \frac{1}{\varepsilon_g} \cdot [\, \alpha \mid x_N \mid + \mid \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} \mid + M_f + (M_g + \mid \hat{g} \mid) \cdot \mid u_c \mid \,] . \tag{9.136}$$

or,

$$\mid u \mid \leq (1 + \frac{M_g + \mid \hat{g} \mid}{\varepsilon_g}) \cdot \mid u_c \mid + \frac{1}{\varepsilon_g} \cdot [\, \alpha \mid x_N \mid + \mid \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} \mid + M_f \,] . \tag{9.137}$$

Equation (9.22) with (9.137) and defining $\alpha \triangleq \alpha_N$ gives

$$\begin{aligned} \mid u \mid \leq\ & (1 + \frac{M_g + \mid \hat{g} \mid}{\varepsilon_g}) \cdot \frac{1}{\mid \hat{g} \mid} \cdot \mid [\, \dot{q}_N + \alpha x_N - \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} - \sum_{k=1}^{N} \alpha_k \epsilon_k \,] \mid \\ & + \frac{1}{\varepsilon_g} \cdot [\, \alpha \mid x_N \mid + \mid \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} \mid + M_f \,] \\ \leq\ & (1 + \frac{M_g + \mid \hat{g} \mid}{\varepsilon_g}) \cdot \frac{1}{\varepsilon_g} \cdot [\, \mid \dot{q}_N \mid + \alpha \mid x_N \mid + \mid \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} \mid + \mid \sum_{k=1}^{N} \alpha_k \epsilon_k \mid \,] \\ & + \frac{1}{\varepsilon_g} \cdot [\, \alpha \mid x_N \mid + \mid \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} \mid + M_f \,] \end{aligned} \tag{9.138}$$

Since

$$\begin{cases} \mid \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} \mid \leq \parallel \mathbf{\Theta}(\mathbf{x},0) \parallel \cdot \parallel \overline{\mathbf{Y}} \parallel \leq \sqrt{I} M_{\overline{\mathbf{Y}}} , \\ \mid \sum_{k=1}^{N} \alpha_k \epsilon_k \mid = \mid \mathbf{a}^T\mathbf{e} \mid \leq \parallel \mathbf{a} \parallel \cdot \parallel \mathbf{e} \parallel \leq \parallel \mathbf{a} \parallel \cdot M_{\mathbf{e}} . \end{cases} \tag{9.139}$$

where I is the size of vector $\mathbf{\Theta}$. Using eq.(9.139) in eq.(9.138) yields

$$\begin{aligned} \mid u \mid \leq\ & \frac{1}{\varepsilon_g} \cdot [\, (2 + \frac{M_g + \mid \hat{g} \mid}{\varepsilon_g}) \cdot (\alpha \mid x_N \mid + \sqrt{I} M_{\overline{\mathbf{Y}}}) \\ & + (1 + \frac{M_g + \mid \hat{g} \mid}{\varepsilon_g}) \cdot (\mid \dot{q}_N \mid + \parallel \mathbf{a} \parallel \cdot M_{\mathbf{e}}) + M_f \,] . \end{aligned} \tag{9.140}$$

This completes the proof of Theorem 9.1.4.

9.8.6 Proof of Theorem 9.1.5 – Estimation of $g(\mathbf{x})$ with Self-tuning Scheme

Consider the Lyapunov function candidate

$$V = \frac{1}{2}[\, \mathbf{e}^T\mathbf{P}\mathbf{e} + h\xi^2 + \boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^T\mathbf{H}^{-1}\boldsymbol{\Delta}_{\overline{\mathbf{Y}}} + \gamma^{-1}\Delta_g^2 \,]\,. \tag{9.141}$$

Differentiating both sides of eq.(9.141) yields,

$$\dot{V} = \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{P}\mathbf{e} + \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} + h\xi\dot{\xi} + \boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^T\mathbf{H}^{-1}\dot{\boldsymbol{\Delta}}_{\overline{\mathbf{Y}}} + \gamma^{-1}\Delta_g\dot{\Delta}_g\,. \tag{9.142}$$

Equations (9.14) and (9.30) together with eq.(9.142) gives

$$\begin{aligned}\dot{V} \;=\; & \frac{1}{2}\mathbf{e}^T(\mathbf{A}^T\mathbf{P} + \mathbf{P}^T\mathbf{A})\mathbf{e} - \alpha h\xi^2 + \boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^T[\, \mathbf{H}^{-1}\dot{\boldsymbol{\Delta}}_{\overline{\mathbf{Y}}} - \boldsymbol{\Theta}(\mathbf{x},0)(\mathbf{B}^T\mathbf{P}\mathbf{e}) + \boldsymbol{\Theta}(\mathbf{x},u)h\xi \,] \\ & + \Delta_g[\, \gamma^{-1}\dot{\Delta}_g - (\mathbf{B}^T\mathbf{P}\mathbf{e})u_s \,] - r(\mathbf{x},0,\boldsymbol{\Theta},\overline{\mathbf{Y}}^*)(\mathbf{B}^T\mathbf{P}\mathbf{e}) + r(\mathbf{x},u,\boldsymbol{\Theta},\overline{\mathbf{Y}}^*)h\xi \\ & - S_u g(\mathbf{B}^T\mathbf{P}\mathbf{e})u_s\,. \end{aligned} \tag{9.143}$$

Using eq.(9.33) in eq.(9.143), and noting that

$$\begin{cases} \dot{\boldsymbol{\Delta}}_{\overline{\mathbf{Y}}} &= \dot{\overline{\mathbf{Y}}}\,. \\ \dot{\Delta}_g &= \dot{g}\,. \\ S_u g(\mathbf{B}^T\mathbf{P}\mathbf{e})u_s &\geq 0. \end{cases} \tag{9.144}$$

we have

$$\begin{aligned}\dot{V} \;\leq\; & -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} - \alpha h\xi^2 + \boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^T[\, \mathbf{H}^{-1}\dot{\overline{\mathbf{Y}}} - \boldsymbol{\Theta}(\mathbf{x},0)(\mathbf{B}^T\mathbf{P}\mathbf{e}) + \boldsymbol{\Theta}(\mathbf{x},u)h\xi \,] \\ & + \Delta_g[\, \gamma^{-1}\dot{g} - (\mathbf{B}^T\mathbf{P}\mathbf{e})u_s \,] \\ & - r(\mathbf{x},0,\boldsymbol{\Theta},\overline{\mathbf{Y}}^*)(\mathbf{B}^T\mathbf{P}\mathbf{e}) + r(\mathbf{x},u,\boldsymbol{\Theta},\overline{\mathbf{Y}}^*)h\xi\,. \end{aligned} \tag{9.145}$$

Substituting eq.(9.32) in eq.(9.145) yields

$$\begin{aligned}\dot{V} \;\leq\; & -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} - \alpha h\xi^2 - S\beta\boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^T\overline{\mathbf{Y}} + \Delta_g[\, \gamma^{-1}\dot{g} - (\mathbf{B}^T\mathbf{P}\mathbf{e})u_s \,] \\ & - r(\mathbf{x},0,\boldsymbol{\Theta},\overline{\mathbf{Y}}^*)(\mathbf{B}^T\mathbf{P}\mathbf{e}) + r(\mathbf{x},u,\boldsymbol{\Theta},\overline{\mathbf{Y}}^*)h\xi\,. \end{aligned} \tag{9.146}$$

After some manipulations, we obtain

$$\boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^{T}\overline{\mathbf{Y}} = \frac{1}{2} \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}} \|^2 + \frac{1}{2} \| \overline{\mathbf{Y}} \|^2 - \frac{1}{2} \| \overline{\mathbf{Y}}^{*} \|^2 \ . \tag{9.147}$$

We know that $\| \overline{\mathbf{Y}}^{*} \| \leq M_{\overline{\mathbf{Y}}}$, and for $S = 1$, $\| \overline{\mathbf{Y}} \| \geq M_{\overline{\mathbf{Y}}}$, therefore,

$$S\beta\boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^{T}\overline{\mathbf{Y}} \geq \frac{1}{2} S\beta \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}} \|^2 \geq 0 \ . \tag{9.148}$$

Since the eigenvalues of a symmetric positive definite matrix are positive[66], the eigenvalues of $\mathbf{Q}$ are positive. Denoting $\lambda_{\min}(\mathbf{Q})$ as the minimum eigenvalue of $\mathbf{Q}$, then

$$\mathbf{e}^{T}\mathbf{Q}\mathbf{e} \geq \lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 \geq 0 \ . \tag{9.149}$$

Substituting eqs.(9.148) and (9.149) into eq.(9.146) yields

$$\begin{aligned} \dot{V} \ \leq \ & -\frac{1}{2}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 - \alpha h\xi^2 - \frac{1}{2} S\beta \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}} \|^2 + \Delta_g [\ \cdot^{-1}\dot{g} - (\mathbf{B}^{T}\mathbf{P}\mathbf{e})u \ ] \\ & - r(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*})(\mathbf{B}^{T}\mathbf{P}\mathbf{e}) + r(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*})h\xi \ . \end{aligned} \tag{9.150}$$

Reorganizing individual terms in eq.(9.150) gives

$$\begin{aligned} -\alpha h\xi^2 + r(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*})h\xi \ &= \ -\frac{1}{2}\alpha h\xi^2 - \frac{1}{2}[\ (\alpha h)^{\frac{1}{2}}\xi - (\frac{h}{\alpha})^{\frac{1}{2}} r(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*})\ ]^2 \\ & \quad + \frac{h}{2\alpha} r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*}) \\ & \leq \ -\frac{1}{2}\alpha h\xi^2 + \frac{h}{2\alpha} r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*}) \ ; \end{aligned} \tag{9.151}$$

$$\begin{aligned} & -\frac{1}{2}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 - r(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*})(\mathbf{B}^{T}\mathbf{P}\mathbf{e}) \\ & \quad = -\frac{1}{4}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 - \frac{1}{4}\lambda_{\min}(\mathbf{Q}) \| [\ \mathbf{e} + 2\frac{r(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*})}{\lambda_{\min}(\mathbf{Q})}(\mathbf{P}^{T}\mathbf{B})\ ] \|^2 \\ & \qquad + \frac{r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*})}{\lambda_{\min}(\mathbf{Q})} \| (\mathbf{P}^{T}\mathbf{B}) \|^2 \\ & \quad \leq -\frac{1}{4}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 + \frac{r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^{*})}{\lambda_{\min}(\mathbf{Q})} \| \mathbf{P}^{T}\mathbf{B} \|^2 \ . \end{aligned} \tag{9.152}$$

Substituting eqs.(9.151-9.152) into eq.(9.150) and recognizing that $-\frac{1}{2}S\mathcal{J} \parallel \mathbf{\Delta}_{\overline{\mathbf{Y}}} \parallel^2 \leq 0$ results in

$$\dot{V} \leq -\frac{1}{4}\lambda_{\min}(\mathbf{Q}) \parallel \mathbf{e} \parallel^2 - \frac{1}{2}\alpha h\xi^2 + \Delta_g[\,\gamma^{-1}\dot{\hat{g}} - (\mathbf{B}^T\mathbf{P}\mathbf{e})u_c\,] + \frac{\parallel \mathbf{P}^T\mathbf{B} \parallel^2}{\lambda_{\min}(\mathbf{Q})} r^2(\mathbf{x}, 0, \mathbf{\Theta}, \overline{\mathbf{Y}}) + \frac{h}{2\alpha} r^2(\mathbf{x}, u, \mathbf{\Theta}, \overline{\mathbf{Y}}) \,. \tag{9.153}$$

Referring to eq.(9.36),

$$\Delta_g[\,\gamma^{-1}\dot{\hat{g}} - (\mathbf{B}^T\mathbf{P}\mathbf{e})u_c\,] = S_g\mathcal{J}_g\Delta_g\dot{\hat{g}} \,. \tag{9.154}$$

where

$$\Delta_g\dot{\hat{g}} = \frac{1}{2}(\dot{g} - \dot{\hat{g}})^2 + \frac{1}{2}\dot{\hat{g}}^2 - \frac{1}{2}\dot{g}^2 = \frac{1}{2}\Delta_g^2 + \frac{1}{2}\dot{\hat{g}}^2 - \frac{1}{2}\dot{g}^2 \,. \tag{9.155}$$

and to eq.(9.37), we consider the following situations:

- For $S_g = 0$,

$$S_g\mathcal{J}_g\Delta_g\dot{\hat{g}} = 0 \,. \tag{9.156}$$

- For $S_g = -1$, $\mid \dot{\hat{g}} \mid \geq M_g$. Since $\mid \dot{g} \mid \leq M_g$, referring to eq.(9.155) gives

$$\Delta_g\dot{\hat{g}} \geq \frac{1}{2}\Delta_g^2 \,. \tag{9.157}$$

or,

$$S_g\mathcal{J}_g\Delta_g\dot{\hat{g}} = -\mathcal{J}_g\Delta_g\dot{\hat{g}} \leq -\frac{1}{2}\mathcal{J}_g\Delta_g^2 \,. \tag{9.158}$$

- For $S_g = 1$, $\mid \dot{\hat{g}} \mid = \varepsilon_g$. Since $\mid \dot{g} \mid \geq \varepsilon_g$, referring to eq.(9.155) gives

$$\Delta_g\dot{\hat{g}} \leq \frac{1}{2}\Delta_g^2 \,. \tag{9.159}$$

or,

$$S_g\mathcal{J}_g\Delta_g\dot{\hat{g}} = \mathcal{J}_g\Delta_g\dot{\hat{g}} \leq \frac{1}{2}\mathcal{J}_g\Delta_g^2 \,. \tag{9.160}$$

From eqs.(9.156), (9.158), and (9.160), it is concluded that

$$S_g \beta_g \Delta_g \dot{g} \leq \frac{1}{2} \beta_g \Delta_g^2 . \tag{9.161}$$

Using eq.(9.161) in (9.153) yields

$$\begin{aligned} \dot{V} \leq & -\frac{1}{4}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 - \frac{1}{2}\alpha h \xi^2 + \frac{1}{2}\beta_g \Delta_g^2 \\ & + \frac{\| \mathbf{P}^T \mathbf{B} \|^2}{\lambda_{\min}(\mathbf{Q})} r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \frac{h}{2\alpha} r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) . \end{aligned} \tag{9.162}$$

Let

$$\zeta \triangleq \min\{\frac{\lambda_{\min}(\mathbf{Q})}{4} , \frac{\alpha h}{2}\} . \tag{9.163}$$

$$\eta \triangleq \max\{\frac{\| \mathbf{P}^T \mathbf{B} \|^2}{\zeta \lambda_{\min}(\mathbf{Q})} , \frac{h}{2\zeta\alpha} , \frac{\beta_g}{2\zeta}\} . \tag{9.164}$$

Then

$$\dot{V} \leq -\zeta(\| \mathbf{e} \|^2 + \xi^2) + \zeta\eta\{ r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2 \} . \tag{9.165}$$

Integrating both sides of eq.(9.165) yields

$$\begin{aligned} V(t) - V(0) \leq & -\zeta \int_0^t \| \mathbf{e}^2(\tau) \| d\tau - \zeta \int_0^t \xi^2(\tau) d\tau \\ & + \zeta\eta \int_0^t \{ r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2 \} d\tau \end{aligned} \tag{9.166}$$

Let

$$b \triangleq \frac{1}{\zeta} \sup_{t \geq 0}\{V(0) - V(t)\} . \tag{9.167}$$

then

$$\begin{aligned} & \int_0^t \| \mathbf{e}(\tau) \|^2 d\tau + \int_0^t \xi^2(\tau) d\tau \\ & \leq b + \eta \int_0^t \{ r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2 \} d\tau . \end{aligned} \tag{9.168}$$

or

$$\int_0^t \| \mathbf{e}(\tau) \|^2 \, d\tau \leq b + \eta \int_0^t [\, r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2 \,] d\tau. \tag{9.169}$$

$$\int_0^t \xi^2(\tau) d\tau \leq b + \eta \int_0^t [\, r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2 \,] d\tau. \tag{9.170}$$

This completes the proof of Theorem 9.1.5 for $g(\mathbf{x})$ being estimated with the self-tuning scheme.

9.8.7 Proof of Theorem 9.1.5 – Estimation of $g(\mathbf{x})$ with FLS

(1) $| \hat{g} | > \varepsilon_g$:

Let Δ_g be the modeling error of the FLS, i.e.,

$$\Delta_g(\mathbf{x}, \boldsymbol{\Psi}, \overline{\mathbf{G}}) \triangleq \hat{g} - g(\mathbf{x}) = \boldsymbol{\Psi}^T \overline{\mathbf{G}} - g(\mathbf{x}) \, . \tag{9.171}$$

and let $\overline{\mathbf{G}}^*$ be the optimal parameter vector that minimizes $\Delta_g(\mathbf{x}, \boldsymbol{\Psi}, \overline{\mathbf{G}}^*)$. Formally,

$$\overline{\mathbf{G}}^* \triangleq arg \min_{\overline{\mathbf{G}}'} \{ \sup_{\mathbf{x} \in X} | \Delta_g(\mathbf{x}, \boldsymbol{\Psi}, \overline{\mathbf{G}}') | \} \, . \tag{9.172}$$

where

$$\overline{\mathbf{G}}' \triangleq \{ \overline{\mathbf{G}} : \| \overline{\mathbf{G}} \| \leq M_{\overline{\mathbf{G}}} \} \, . \tag{9.173}$$

The existence of such an optimal parameter vector, $\overline{\mathbf{G}}^*$, is justified by Lemma 8.1. $M_{\overline{\mathbf{G}}}$ is a positive constant that bounds $\| \overline{\mathbf{G}} \|$. Let

$$\hat{g}^* \triangleq \boldsymbol{\Psi}^T \overline{\mathbf{G}}^* \, . \tag{9.174}$$

Consider eq.(9.30).

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}[\, -\boldsymbol{\Theta}^T(\mathbf{x},0)\boldsymbol{\Delta}_{\overline{\mathbf{Y}}} - \Delta_g u_c - r(\mathbf{x},0,\boldsymbol{\Theta},\overline{\mathbf{Y}}^*) - S_u g u_s \,]\,. \tag{9.175}$$

Since

$$\begin{aligned} \Delta_g u_c &= (\hat{g} - g)u_c \\ &= (\hat{g} - \hat{g}^*)u_c + (\hat{g}^* - g)u_c \\ &= \boldsymbol{\Psi}^T(\overline{\mathbf{G}} - \overline{\mathbf{G}}^*)u_c + (\boldsymbol{\Psi}^T\overline{\mathbf{G}}^* - g)u_c\,. \end{aligned} \tag{9.176}$$

and referring to eq.(9.171) and letting

$$\boldsymbol{\Delta}_{\overline{\mathbf{G}}} \triangleq \overline{\mathbf{G}} - \overline{\mathbf{G}}^*\,. \tag{9.177}$$

we have

$$\Delta_g u_c = \boldsymbol{\Psi}^T\boldsymbol{\Delta}_{\overline{\mathbf{G}}}u_c + \Delta_g(\mathbf{x},\boldsymbol{\Psi},\overline{\mathbf{G}}^*)u_c\,. \tag{9.178}$$

Equation (9.178) in eq.(9.175) yields

$$\begin{aligned} \dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}[\, &-\boldsymbol{\Theta}^T(\mathbf{x},0)\boldsymbol{\Delta}_{\overline{\mathbf{Y}}} - \boldsymbol{\Psi}^T\boldsymbol{\Delta}_{\overline{\mathbf{G}}}u_c - r(\mathbf{x},0,\boldsymbol{\Theta},\overline{\mathbf{Y}}^*) - \Delta_g(\mathbf{x},\boldsymbol{\Psi},\overline{\mathbf{G}}^*)u_c \\ &- S_u g u_s \,]\,. \end{aligned} \tag{9.179}$$

Consider the Lyapunov function candidate

$$V = \frac{1}{2}[\, \mathbf{e}^T\mathbf{P}\mathbf{e} + h\xi^2 + \boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^T\mathbf{H}^{-1}\boldsymbol{\Delta}_{\overline{\mathbf{Y}}} + \gamma^{-1}\boldsymbol{\Delta}_{\overline{\mathbf{G}}}^T\boldsymbol{\Delta}_{\overline{\mathbf{G}}} \,]\,. \tag{9.180}$$

Differentiating both sides of eq.(9.180) yields.

$$\dot{V} = \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{P}\mathbf{e} + \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} + h\xi\dot{\xi} + \boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^T\mathbf{H}^{-1}\dot{\boldsymbol{\Delta}}_{\overline{\mathbf{Y}}} + \gamma^{-1}\boldsymbol{\Delta}_{\overline{\mathbf{G}}}^T\dot{\boldsymbol{\Delta}}_{\overline{\mathbf{G}}}\,. \tag{9.181}$$

In the current situation, $\{S_g,\ \mathcal{J}_g\}^T = \{S_g^U,\ \mathcal{J}_g^U\}^T$. Using eqs.(9.14) and (9.179) in (9.181) yields

$$\dot{V} = \frac{1}{2}\mathbf{e}^T(\mathbf{A}^T\mathbf{P} + \mathbf{P}^T\mathbf{A})\mathbf{e} - \alpha h\xi^2$$

$$
\begin{aligned}
&+ \boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^{T}[\ \mathbf{H}^{-1}\dot{\boldsymbol{\Delta}}_{\overline{\mathbf{Y}}} - \boldsymbol{\Theta}(\mathbf{x},0)(\mathbf{B}^{T}\mathbf{P}\mathbf{e}) + \boldsymbol{\Theta}(\mathbf{x},u)h\xi\ ] \\
&+ \boldsymbol{\Delta}_{\overline{\mathbf{G}}}^{T}[\ \gamma^{-1}\dot{\boldsymbol{\Delta}}_{\overline{\mathbf{G}}} - \boldsymbol{\Psi}(\mathbf{B}^{T}\mathbf{P}\mathbf{e})u_{c}\ ] - S_{u}g(\mathbf{B}^{T}\mathbf{P}\mathbf{e})u_{s} \\
&- r(\mathbf{x},0,\boldsymbol{\Theta},\overline{\mathbf{Y}}^{*})(\mathbf{B}^{T}\mathbf{P}\mathbf{e}) + r(\mathbf{x},u,\boldsymbol{\Theta},\overline{\mathbf{Y}}^{*})h\xi \\
&- \Delta_{g}(\mathbf{x},\boldsymbol{\Psi},\overline{\mathbf{G}}^{*})(\mathbf{B}^{T}\mathbf{P}\mathbf{e})u_{c}\ . \qquad (9.182)
\end{aligned}
$$

Using eq.(9.33) in eq.(9.182), and noting that

$$
\begin{cases}
\dot{\boldsymbol{\Delta}}_{\overline{\mathbf{Y}}} & = \dot{\overline{\mathbf{Y}}}\ . \\
\dot{\boldsymbol{\Delta}}_{\overline{\mathbf{G}}} & = \dot{\overline{\mathbf{G}}}\ . \\
S_{u}g(\mathbf{B}^{T}\mathbf{P}\mathbf{e})u_{s} & \geq 0.
\end{cases} \qquad (9.183)
$$

we have

$$
\begin{aligned}
\dot{V} \leq\ & -\frac{1}{2}\mathbf{e}^{T}\mathbf{Q}\mathbf{e} - \alpha h\xi^{2} + \boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^{T}[\ \mathbf{H}^{-1}\dot{\overline{\mathbf{Y}}} - \boldsymbol{\Theta}(\mathbf{x},0)(\mathbf{B}^{T}\mathbf{P}\mathbf{e}) + \boldsymbol{\Theta}(\mathbf{x},u)h\xi\ ] \\
&+ \boldsymbol{\Delta}_{\overline{\mathbf{G}}}^{T}[\ \gamma^{-1}\dot{\overline{\mathbf{G}}} - \boldsymbol{\Psi}(\mathbf{B}^{T}\mathbf{P}\mathbf{e})u_{c}\ ] - S_{u}g(\mathbf{B}^{T}\mathbf{P}\mathbf{e})u_{s} \\
&- r(\mathbf{x},0,\boldsymbol{\Theta},\overline{\mathbf{Y}}^{*})(\mathbf{B}^{T}\mathbf{P}\mathbf{e}) - r(\mathbf{x},u,\boldsymbol{\Theta},\overline{\mathbf{Y}}^{*})h\xi \\
&- \Delta_{g}(\mathbf{x},\boldsymbol{\Psi},\overline{\mathbf{G}}^{*})(\mathbf{B}^{T}\mathbf{P}\mathbf{e})u_{c}\ . \qquad (9.184)
\end{aligned}
$$

Substitution of (9.32) and (9.43) in eq.(9.184) yields

$$
\begin{aligned}
\dot{V} \leq\ & -\frac{1}{2}\mathbf{e}^{T}\mathbf{Q}\mathbf{e} - \alpha h\xi^{2} - S\beta\boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^{T}\overline{\mathbf{Y}} - S_{g}^{U}\beta_{g}^{U}\boldsymbol{\Delta}_{\overline{\mathbf{G}}}^{T}\overline{\mathbf{G}} \\
&- r(\mathbf{x},0,\boldsymbol{\Theta},\overline{\mathbf{Y}}^{*})(\mathbf{B}^{T}\mathbf{P}\mathbf{e}) + r(\mathbf{x},u,\boldsymbol{\Theta},\overline{\mathbf{Y}}^{*})h\xi \\
&- \Delta_{g}(\mathbf{x},\boldsymbol{\Psi},\overline{\mathbf{G}}^{*})(\mathbf{B}^{T}\mathbf{P}\mathbf{e})u_{c}\ . \qquad (9.185)
\end{aligned}
$$

$$
\boldsymbol{\Delta}_{\overline{\mathbf{G}}}^{T}\overline{\mathbf{G}} = \frac{1}{2}\parallel\boldsymbol{\Delta}_{\overline{\mathbf{G}}}\parallel^{2} + \frac{1}{2}\parallel\overline{\mathbf{G}}\parallel^{2} - \frac{1}{2}\parallel\overline{\mathbf{G}}^{*}\parallel^{2}\ . \qquad (9.186)
$$

- For $S_{g}^{U} = 0$,

$$
-S_{g}^{U}\beta_{g}^{U}\boldsymbol{\Delta}_{\overline{\mathbf{G}}}^{T}\overline{\mathbf{G}} = 0\ . \qquad (9.187)
$$

- For $S_{g}^{U} = 1$, $\parallel\overline{\mathbf{G}}\parallel \geq M_{\overline{\mathbf{G}}}$, but $\parallel\overline{\mathbf{G}}^{*}\parallel \leq M_{\overline{\mathbf{G}}}$. Therefore,

$$
-S_{g}^{U}\beta_{g}^{U}\boldsymbol{\Delta}_{\overline{\mathbf{G}}}^{T}\overline{\mathbf{G}} \leq -\frac{1}{2}\beta_{g}^{U}\parallel\boldsymbol{\Delta}_{\overline{\mathbf{G}}}\parallel^{2}\ . \qquad (9.188)
$$

From eqs.(9.187) and (9.188), we conclude that

$$-S_g^U \beta_g^U \Delta_{\overline{\mathbf{G}}}^T \overline{\mathbf{G}} \leq -\frac{1}{2} S_g^U \beta_g^U \| \Delta_{\overline{\mathbf{G}}} \|^2 \quad . \tag{9.189}$$

Using eqs.(9.189) and (9.148-9.149) in eq.(9.185) yields

$$\begin{aligned} \dot{V} \leq & -\frac{1}{2}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 - \alpha h \xi^2 - \frac{1}{2} S\beta \| \Delta_{\overline{\mathbf{Y}}} \|^2 - \frac{1}{2} S_g^U \beta_g^U \| \Delta_{\overline{\mathbf{G}}} \|^2 \\ & - r(\mathbf{x}, 0, \Theta, \overline{\mathbf{Y}}^*)(\mathbf{B}^T \mathbf{P} \mathbf{e}) + r(\mathbf{x}, u, \Theta, \overline{\mathbf{Y}}^*) h \xi \\ & - \Delta_g(\mathbf{x}, \Psi, \overline{\mathbf{G}}^*)(\mathbf{B}^T \mathbf{P} \mathbf{e}) u_s \quad . \end{aligned} \tag{9.190}$$

Individual terms in eq.(9.190) can be reorganized as:

$$\begin{aligned} & -\frac{1}{2}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 - r(\mathbf{x}, 0, \Theta, \overline{\mathbf{Y}}^*)(\mathbf{B}^T \mathbf{P} \mathbf{e}) - \Delta_g(\mathbf{x}, \Psi, \overline{\mathbf{G}}^*)(\mathbf{B}^T \mathbf{P} \mathbf{e}) u_s \\ & = -\frac{1}{4}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 - \frac{1}{8}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} + 4\frac{r(\mathbf{x}, 0, \Theta, \overline{\mathbf{Y}}^*)}{\lambda_{\min}(\mathbf{Q})}(\mathbf{P}^T \mathbf{B}) \|^2 \\ & \quad + 2\frac{r^2(\mathbf{x}, 0, \Theta, \overline{\mathbf{Y}}^*)}{\lambda_{\min}(\mathbf{Q})} \| (\mathbf{P}^T \mathbf{B}) \|^2 + 2\frac{\Delta_g^2(\mathbf{x}, \Psi, \overline{\mathbf{G}}^*) u_s^2}{\lambda_{\min}(\mathbf{Q})} \| (\mathbf{P}^T \mathbf{B}) \|^2 \\ & \quad - \frac{1}{8}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} + 4\frac{\Delta_g(\mathbf{x}, \Psi, \overline{\mathbf{G}}^*) u_s}{\lambda_{\min}(\mathbf{Q})}(\mathbf{P}^T \mathbf{B}) \|^2 \\ & \leq -\frac{1}{4}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 + 2\frac{r^2(\mathbf{x}, 0, \Theta, \overline{\mathbf{Y}}^*)}{\lambda_{\min}(\mathbf{Q})} \| \mathbf{P}^T \mathbf{B} \|^2 \\ & \quad + 2\frac{\Delta_g^2(\mathbf{x}, \Psi, \overline{\mathbf{G}}^*) u_s^2}{\lambda_{\min}(\mathbf{Q})} \| \mathbf{P}^T \mathbf{B} \|^2 \quad . \end{aligned} \tag{9.191}$$

Using eqs.(9.151) and (9.191) in eq.(9.190), and noting that

$$\begin{cases} -\frac{1}{2} S\beta \| \Delta_{\overline{\mathbf{Y}}} \|^2 & \leq 0 \ . \\ -\frac{1}{2} S_g^U \beta_g^U \| \Delta_{\overline{\mathbf{G}}} \|^2 & \leq 0 \ . \end{cases} \tag{9.192}$$

then

$$\begin{aligned} \dot{V} \leq & -\frac{1}{4}\lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 - \frac{1}{2}\alpha h \xi^2 \\ & + 2\frac{\| \mathbf{P}^T \mathbf{B} \|^2}{\lambda_{\min}(\mathbf{Q})} [\, r^2(\mathbf{x}, 0, \Theta, \overline{\mathbf{Y}}^*) + \Delta_g^2(\mathbf{x}, \Psi, \overline{\mathbf{G}}^*) u_s^2 \,] \\ & + \frac{h}{2\alpha} r^2(\mathbf{x}, u, \Theta, \overline{\mathbf{G}}^*) \ . \end{aligned} \tag{9.193}$$

Since

$$u_c \triangleq \frac{1}{\hat{g}}[\ \dot{q}_N + \alpha q_N - \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}} - \sum_{k=1}^{N-1}\alpha_k\epsilon_k\ ]\ . \tag{9.194}$$

then

$$\begin{aligned} |\ u_c\ | &\leq \frac{1}{\varepsilon_g}[\ |\ \dot{q}_N\ | + \alpha\ |\ x_N\ | + |\ \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}}\ | + |\ \sum_{k=1}^{N}\alpha_k\epsilon_k\ |\ ] \\ &\leq \frac{1}{\varepsilon_g}[\ M_{\dot{q}N} + \alpha M_{xN} + \|\ \mathbf{\Theta}^T(\mathbf{x},0)\ \|\cdot\|\ \overline{\mathbf{Y}}\ \| + \|\ \mathbf{a}\ \|\cdot\|\ \mathbf{e}\ \|\ ] \\ &\leq \frac{1}{\varepsilon_g}[\ M_{\dot{q}N} + \alpha M_{xN} - \sqrt{l}M_{\overline{\mathbf{Y}}} + \|\ \mathbf{a}\ \| M_{\mathbf{e}}\ ]\ . \end{aligned} \tag{9.195}$$

where

$$\begin{cases} M_{\dot{q}N} \triangleq \sup_{t\geq 0}\{|\ \dot{q}_N\ |\}\ . \\ M_{xN} \triangleq \sup_{t\geq 0}\{|\ x_N\ |\}\ . \end{cases} \tag{9.196}$$

Let

$$M_{u_c} \triangleq \frac{1}{\varepsilon_g}[\ M_{\dot{q}N} + \alpha M_{xN} + \sqrt{l}M_{\overline{\mathbf{Y}}} + \|\ \mathbf{a}\ \| M_{\mathbf{e}}\ ]\ . \tag{9.197}$$

we have

$$|\ u_c\ | \leq M_{u_c}\ . \tag{9.198}$$

Let

$$\zeta \triangleq \min\{\frac{\lambda_{\min}(\mathbf{Q})}{4}\ ,\ \frac{\alpha h}{2}\}\ . \tag{9.199}$$

$$\eta \triangleq \max\{2\frac{\|\ \mathbf{P}^T\mathbf{B}\ \|^2}{\zeta\lambda_{\min}(\mathbf{Q})}\ ,\ 2\frac{\|\ \mathbf{P}^T\mathbf{B}\ \|^2}{\zeta\lambda_{\min}(\mathbf{Q})}M_{u_c}^2\ ,\ \frac{h}{2\zeta\alpha}\}\ . \tag{9.200}$$

Then

$$\begin{aligned} \dot{V} \leq\ & -\zeta(\|\ \mathbf{e}\ \|^2 + \xi^2) \\ & + \zeta\eta[\ r^2(\mathbf{x},0,\mathbf{\Theta},\overline{\mathbf{Y}}^*) + r^2(\mathbf{x},u_c,\mathbf{\Theta},\overline{\mathbf{Y}}^*) + \Delta_g^2(\mathbf{x},\mathbf{\Psi},\overline{\mathbf{G}}^*)\ ]\ . \end{aligned} \tag{9.201}$$

Integrating both sides of eq.(9.201) yields

$$V(t) - V(0) \leq -\zeta \int_0^t \| \mathbf{e}^2(\tau) \| d\tau - \zeta \int_0^t \xi^2(\tau) d\tau + \zeta\eta \int_0^t [\, r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2(\mathbf{x}, \boldsymbol{\Psi}, \overline{\mathbf{G}}^*) \,] d\tau \,. \tag{9.202}$$

Let

$$b \triangleq \frac{1}{\zeta} \sup_{t \geq 0} \{ V(0) - V(t) \} \,. \tag{9.203}$$

we have

$$\int_0^t \| \mathbf{e}(\tau) \|^2 d\tau + \int_0^t \xi^2(\tau) d\tau \leq b + \eta \int_0^t [\, r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2(\mathbf{x}, \boldsymbol{\Psi}, \overline{\mathbf{G}}^*) \,] d\tau \tag{9.204}$$

or

$$\int_0^t \| \mathbf{e}(\tau) \|^2 d\tau \leq b + \eta \int_0^t [\, r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2(\mathbf{x}, \boldsymbol{\Psi}, \overline{\mathbf{G}}^*) \,] d\tau. \tag{9.205}$$

$$\int_0^t \xi^2(\tau) d\tau \leq b + \eta \int_0^t [\, r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) - r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2(\mathbf{x}, \boldsymbol{\Psi}, \overline{\mathbf{G}}^*) \,] d\tau. \tag{9.206}$$

(2) $|\dot{g}| = \varepsilon_g$:

Consider the Lyapunov function candidate

$$V = \frac{1}{2} [\, \mathbf{e}^T \mathbf{P} \mathbf{e} + h\xi^2 + \boldsymbol{\Delta}_{\overline{\mathbf{Y}}}^T \mathbf{H}^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}} + \gamma^{-1} (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \Delta_g^2 \,] \,. \tag{9.207}$$

By mathematical manipulations similar to those of eqs.(9.142-9.153) we have

$$\dot{V} \leq -\frac{1}{4} \lambda_{\min}(\mathbf{Q}) \| \mathbf{e} \|^2 - \frac{1}{2} \alpha h \xi^2 + \Delta_g [\, \gamma^{-1} (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \dot{g} - (\mathbf{B}^T \mathbf{P} \mathbf{e}) u_s \,] + \frac{\| \mathbf{P}^T \mathbf{B} \|^2}{\lambda_{\min}(\mathbf{Q})} r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \frac{h}{2\alpha} r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) \,. \tag{9.208}$$

Considering $\dot{\hat{g}} = \mathbf{\Psi}^T \dot{\overline{\mathbf{G}}}$, and using eqs.(9.43-9.44) gives

$$\begin{aligned} \Delta_g[\ \gamma^{-1}(\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\dot{\hat{g}} - (\mathbf{B}^T\mathbf{P}\mathbf{e})u_c\ ] &= S_g^L \beta_g^L (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\Delta_g \mathbf{\Psi}^T \overline{\mathbf{G}} \\ &= S_g^L \beta_g^L (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\Delta_g \hat{g}\ . \end{aligned} \tag{9.209}$$

Since

$$\begin{cases} \Delta_g \hat{g} &= \frac{1}{2}\Delta_g^2 + \frac{1}{2}\hat{g}^2 - \frac{1}{2}g^2\ . \\ |\ \hat{g}\ | &= \varepsilon_g\ . \\ |\ g\ | &\geq \varepsilon_g\ . \end{cases} \tag{9.210}$$

we have

$$\Delta_g \hat{g} \leq \frac{1}{2}\Delta_g^2\ . \tag{9.211}$$

Using eq.(9.211) in (9.209), and considering both $S_g^L = 0$ and $S_g^L = 1$, we have

$$\Delta_g[\ \gamma^{-1}(\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\dot{\hat{g}} - (\mathbf{B}^T\mathbf{P}\mathbf{e})u_c\ ] \leq \frac{1}{2}\beta_g^L (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\Delta_g^2\ . \tag{9.212}$$

Equation (9.212) in (9.208) gives

$$\begin{aligned} \dot{V} \leq\ & -\frac{1}{4}\lambda_{\min}(\mathbf{Q})\ \|\ \mathbf{e}\ \|^2 - \frac{1}{2}\alpha h \xi^2 + \frac{1}{2}\beta_g^L (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\Delta_g^2 \\ & + \frac{\|\ \mathbf{P}^T\mathbf{B}\ \|^2}{\lambda_{\min}(\mathbf{Q})} r^2(\mathbf{x}, 0, \mathbf{\Theta}, \overline{\mathbf{Y}}^*) + \frac{h}{2\alpha} r^2(\mathbf{x}, u, \mathbf{\Theta}, \overline{\mathbf{Y}}^*)\ . \end{aligned} \tag{9.213}$$

Let

$$\zeta \triangleq \min\{\frac{\lambda_{\min}(\mathbf{Q})}{4}\ ,\ \frac{\alpha h}{2}\}\ . \tag{9.214}$$

$$\eta \triangleq \max\{\frac{\|\ \mathbf{P}^T\mathbf{B}\ \|^2}{\zeta\lambda_{\min}(\mathbf{Q})}\ ,\ \frac{h}{2\zeta\alpha}\ ,\ \frac{\beta_g^L (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}}{2\zeta}\}\ . \tag{9.215}$$

Then

$$\dot{V} \leq -\zeta(\|\ \mathbf{e}\ \|^2 + \xi^2) + \zeta\eta[\ r^2(\mathbf{x}, 0, \mathbf{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \mathbf{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2\ ]\ . \tag{9.216}$$

Integrating both sides of eq.(9.216) yields

$$\begin{aligned} V(t) - V(0) \leq\ & -\zeta\int_0^t \|\ \mathbf{e}^2(\tau)\ \|\ d\tau - \zeta\int_0^t \xi^2(\tau) d\tau \\ & + \zeta\eta\int_0^t [\ r^2(\mathbf{x}, 0, \mathbf{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \mathbf{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2\ ] d\tau \end{aligned} \tag{9.217}$$

Let

$$b \triangleq \frac{1}{\zeta} \sup_{t \geq 0} \{V(0) - V(t)\} \ . \tag{9.218}$$

then

$$\int_0^t \| \mathbf{e}(\tau) \|^2 \, d\tau + \int_0^t \xi^2(\tau) d\tau$$
$$\leq b + \eta \int_0^t [r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2] d\tau \ . \tag{9.219}$$

or

$$\int_0^t \| \mathbf{e}(\tau) \|^2 \, d\tau \leq b + \eta \int_0^t [r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2] d\tau. \tag{9.220}$$

$$\int_0^t \xi^2(\tau) d\tau \leq b + \eta \int_0^t [r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) + \Delta_g^2] d\tau. \tag{9.221}$$

This completes the proof of Theorem 9.1.5 for $g(\mathbf{x})$ being estimated with a FLS.

9.8.8 Proof of Theorem 9.1.6

For $r(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) \in \mathcal{L}_2[0, \infty)$ and $\Delta_g \in \mathcal{L}_2[0, \infty)$, we have

$$[\int_0^\infty r^2(\mathbf{x}, u, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) d\tau \ ]^{\frac{1}{2}} < \infty \ . \tag{9.222}$$

$$[\int_0^\infty r^2(\mathbf{x}, 0, \boldsymbol{\Theta}, \overline{\mathbf{Y}}^*) d\tau \ ]^{\frac{1}{2}} < \infty \ . \tag{9.223}$$

$$[\int_0^\infty \Delta_g^2 d\tau \ ]^{\frac{1}{2}} < \infty \ . \tag{9.224}$$

By Theorem 9.1.5,

$$[\int_0^\infty \| \mathbf{e}(\tau) \|^2 \, d\tau \ ]^{\frac{1}{2}} < \infty \ . \tag{9.225}$$

$$[\int_0^\infty \xi^2(\tau)d\tau\ ]^{\frac{1}{2}} < \infty\ . \tag{9.226}$$

Therefore, $\mathbf{e} \in \mathcal{L}_2$, and $\xi \in \mathcal{L}_2$. From Theorem 9.1.3, it is known that $\mathbf{e} \in \mathcal{L}_\infty$, and $\xi \in \mathcal{L}_\infty$. Observing the expressions for $\dot{\xi}$ and $\dot{\mathbf{e}}$, eqs.(9.14) and (9.30) reveals that all the components on the right hand sides of the expressions are bounded. Therefore, $\dot{\xi}$ and $\dot{\mathbf{e}}$ are bounded, that is, $\dot{\mathbf{e}} \in \mathcal{L}_\infty$ and $\dot{\xi} \in \mathcal{L}_\infty$. From Corollary 2.9 in [114], we conclude that

$$\lim_{t\to\infty} \| \mathbf{e} \| = 0\ . \tag{9.227}$$

$$\lim_{t\to\infty} | \xi | = 0\ . \tag{9.228}$$

This completes the proof of Theorem 9.1.6.

9.8.9 Proof of Corollaries 9.1 and 9.2

Consider $\dot{g} \equiv g$ and $\Delta_g \equiv 0$. Corollary 9.1 directly follows Theorem 9.1.1, 9.1.3, 9.1.5, and 9.1.6.

For Corollary 9.2, consider eqs.(9.20) and (9.54) then

$$u = u_c - S_c sgn(\mathbf{B}^T\mathbf{P}\mathbf{e}) \cdot \frac{1}{g} \cdot [\ \alpha\ |\ x_N\ | + |\ \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}}\ | + M_f\ ]\ . \tag{9.229}$$

or,

$$|\ u\ | \le |\ u_c\ | + \frac{1}{|\ g\ |} \cdot [\ \alpha\ |\ x_N\ | + |\ \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}}\ | + M_f\ ]\ . \tag{9.230}$$

Using eq.(9.53) in (9.230) and defining $\alpha \hat{=} \alpha_N$ gives

$$\begin{aligned} |\ u\ | &\le \frac{1}{|\ g\ |} \cdot [\ |\ \dot{q}_N\ | + \alpha\ |\ x_N\ | + |\ \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}}\ | + |\ \sum_{k=1}^{N} \alpha_k e_k\ |\ ] \\ &\quad + \frac{1}{|\ g\ |} \cdot [\ \alpha\ |\ x_N\ | + |\ \mathbf{\Theta}^T(\mathbf{x},0)\overline{\mathbf{Y}}\ | + M_f\ ] \\ &\le \frac{1}{|\ g\ |} \cdot [\ 2(\alpha\ |\ x_N\ | + \sqrt{l}M_{\overline{\mathbf{Y}}}) + |\ \dot{q}_N\ | + \|\ \mathbf{a}\ \| \cdot M_e + M_f\ ]\ . \end{aligned} \tag{9.231}$$

This completes the proof of Corollary 9.2.

9.9 Illustrative Applications

9.9.1 Example 1 – Trajectory Control of a Simple Nonlinear System

Consider the following nonlinear system:

$$\dot{x} = \frac{\cos(x)}{1+x^2} + \frac{2}{1+\cos^2(x)}u \ . \tag{9.232}$$

Our objective is to command the state x to track a desired trajectory, $q(t)$, by using the DFLS based adaptive control scheme developed in previous sections, where this trajectory is

$$q \triangleq 2\sin(t) \ . \tag{9.233}$$

A DFLS identifier, labeled D, is used to identify x, i.e.,

$$y = D(x,u) \ , \tag{9.234}$$

where y represents the identifier's output and is the estimate of x. The tracking error, ϵ, and identification error, ξ, are

$$\begin{cases} \epsilon & \triangleq & x-q \ . \\ \xi & \triangleq & y-x \ . \end{cases} \tag{9.235}$$

The identifier, D, has two inputs, $x \in \mathcal{X} \subset R$, $u \in \mathcal{U} \subset R$, and one output, $y \in \mathcal{Y} \subset R$, where $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{Y}$ are universes of discourse of linguistic variables, x, u and y, respectively. For conciseness of expression, let $\mathbf{z} \triangleq \{x,u\}^T \triangleq \{x_1,x_2\}^T$. In both $\mathcal{X}$ and $\mathcal{U}$, five primary fuzzy sets, A_{1j_1} and A_{2j_2}, $j_1, j_2 = 1,\ldots,5$, are defined. Gaussian type membership functions are used for all the primary fuzzy sets,

$$\mu_{A_{pl}}(x'_p) \triangleq \begin{cases} 1 \ , & if \quad x'_p < \bar{x}_{pl} \ . \\ \exp[-\frac{1}{2}(\frac{x'_p-\bar{x}_{pl}}{\sigma_{pl}})^2] \ , & if \quad x'_p \geq \bar{x}_{pl} \ , \end{cases}$$

$$\mu_{A_{pj_p}}(x'_p) \triangleq \exp[-\frac{1}{2}(\frac{x'_p - \overline{x}_{pj_p}}{\sigma_{pj_p}})^2] \ . \quad j_1, j_2 = 2, 3, 4. \tag{9.236}$$

$$\mu_{A_{p5}}(x'_p) \triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{x'_p - \overline{x}_{p5}}{\sigma_{p5}})^2] \ . & if \quad x'_p < \overline{x}_{p5} \ . \\ 1 \ . & if \quad x'_p \geq \overline{x}_{p5} \ . \end{cases}$$

We set the shape parameters of all the primary fuzzy sets to 0.45, i.e.,

$$\sigma_{pj_p} \triangleq 0.45 \ . \tag{9.237}$$

and define the position parameters, $\overline{x}_{pj_p}$, as,

$$\{\overline{x}_{p1}, \ \overline{x}_{p2}, \ \overline{x}_{p3}, \ \overline{x}_{p4}, \ \overline{x}_{p5}\}^T \triangleq \{-2, \ -1, \ 0, \ 1, \ 2\}^T \ . \tag{9.238}$$

In eq.(9.236), x'_p, $p = 1, 2$, are filtered values of inputs, x_p, that is,

$$\begin{cases} x'_1 = f_1(x_1) \ . \\ x'_2 = f_2(x_2) \ . \end{cases} \tag{9.239}$$

The output of D, y, is defined with the DFLS of eq.(9.4), i.e.,

$$D(\mathbf{z}'): \quad \dot{y} = -\alpha y + \mathbf{\Theta}^T(\mathbf{z}')\overline{\mathbf{Y}} \ . \tag{9.240}$$

In this equation,

$$\begin{cases} \mathbf{\Theta}(\mathbf{z}') \triangleq \{\theta_1(\mathbf{z}'), \ \theta_2(\mathbf{z}'), \ \ldots, \ \theta_{25}(\mathbf{z}')\}^T \ . \\ \overline{\mathbf{Y}} \triangleq \{\overline{y}_1, \ \overline{y}_2, \ \ldots, \ \overline{y}_{25}\}^T \ . \end{cases} \tag{9.241}$$

and

$$\theta_i(\mathbf{z}') = \frac{\prod_{p=1}^{2} \mu_{A_p^i}(x'_p)}{\sum_{i=1}^{25} \prod_{p=1}^{2} \mu_{A_p^i}(x'_p)} \ . \qquad i = 1, \ldots, 25. \tag{9.242}$$

where

$$\begin{cases} A_1^i \in \{A_{1j_1}: \ j_1 = 1, \ldots, 5\} \ . \\ A_2^i \in \{A_{2j_2}: \ j_2 = 1, \ldots, 5\} \ . \end{cases} \tag{9.243}$$

$\overline{\mathbf{Y}}$ is the free parameter vector to be adaptively adjusted with the training law, eq.(9.32).

$$\dot{\overline{\mathbf{Y}}} = -\mathbf{H}[\mathbf{\Theta}(x', u')h\xi - \mathbf{\Theta}(x', 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] - S3\mathbf{H}\overline{\mathbf{Y}} \ . \tag{9.244}$$

In this control system, $\mathbf{A} = [-a]$, $\mathbf{B} = [1]$, and $\mathbf{e} = [\epsilon]$. Let $\mathbf{Q} \triangleq [2a]$, then $\mathbf{P} = [1]$. Further, let $h \triangleq 1$, then

$$\dot{\overline{\mathbf{Y}}} = -\mathbf{H}[\mathbf{\Theta}(x', u')\xi - \mathbf{\Theta}(x', 0)\epsilon] - S3\mathbf{H}\overline{\mathbf{Y}} \ . \tag{9.245}$$

In computer implementation, the derivative is approximated by a difference, i.e.,

$$\dot{\overline{\mathbf{Y}}} \triangleq \frac{\overline{\mathbf{Y}}(kT + T) - \overline{\mathbf{Y}}(kT)}{T} \ , \qquad k = 0, 1, 2, \ldots \tag{9.246}$$

where T is the time incremental step. Use of eq.(9.246) in (9.245) yields

$$\begin{aligned} \overline{\mathbf{Y}}(kT + T) \ = \ & \overline{\mathbf{Y}}(kT) - T\mathbf{H}[\ \mathbf{\Theta}(x'(kT), u'(kT))\xi(kT) - \mathbf{\Theta}(x'(kT), 0)\epsilon(kT) \\ & + S3\overline{\mathbf{Y}}(kT)\ ] \ . \end{aligned} \tag{9.247}$$

The gain matrix, $\mathbf{H}$, is defined as a diagonal matrix with all diagonal elements equal to 4. We set a to 10, the time increment, T, to 0.05 sec, the bound of the parameter vector $M_{\overline{\mathbf{Y}}}$, to 10^3, and the bound of the function f, to $M_f = 1$. The initial values of the elements of the parameter vector, $\overline{\mathbf{Y}}(0)$, are random numbers uniformly distributed in $(-1, 1)$, and assume the system to be controlled is initially at rest, i.e., $x(0) = 0$. Now, we consider the following situations.

9.9.1.A Function $g(x)$ Known

Consider the situation where $g(x)$ is a known function. In this situation, no identifier is required to estimate $g(x)$. The prefilters are defined as

$$\begin{cases} x_1' \ = \ f_1(x_1) \ \triangleq \ x_1 \ . \\ x_2' \ = \ f_2(x_2) \ \triangleq \ x_2 \cdot \frac{1}{25} \ . \end{cases} \tag{9.248}$$

The bound of the tracking error, M_ϵ, is set to 15. The controller output, u, the identifier output, y, and the controlled trajectory of system output, x, are shown in fig.9.1(a-c), respectively. The adaptation process ends at $t = 20$ sec.

It is seen that the training process for the DFLS identifier converges very well, and the the control strategy is very effective. After the adaptation process ends the identifier predicts the system state very well, and the system output follows the desired trajectory satisfactorily. In this case, the tracking error, ϵ, is always within its bound, M_ϵ, and therefore, the supervisory control is not triggered. The tracking error for this case is shown in fig.9.3(a).

With the bound of the tracking error, M_ϵ, set to 6, the controller output, u, the identifier output, y, and the controlled trajectory of system output, x, are shown in fig.9.2(a-c), respectively, where the adaptation process starts at $t = 0.5$ sec, and ends at $t = 20$ sec. The tracking error for this case is shown in fig.9.3(b). It is observed that in this situation the tracking error is generally contained within its desired bound, and whenever the tracking error is detected exceeding its bound the supervisory control action is triggered and the error is pulled back within the bound. Nevertheless this is achieved with extraordinary control efforts, as observed from fig.9.2(a). Since the supervisory control is to prevent the system states from drifting to infinity, it is usually very large. Therefore, unless absolutely necessary, this action should generally be avoided in practice by careful off-line designs.

9.9.1.B Function $g(x)$ Unknown and Estimated with Self-Tuning Scheme

Consider the situation where $g(x)$ is an unknown function, and we use the self-tuning scheme to estimate its value. We set the prefilters as

$$\begin{cases} x_1' &= f_1(x_1) \triangleq x_1 \ . \\ x_2' &= f_2(x_2) \triangleq x_2 \cdot \frac{1}{25} \ . \end{cases} \tag{9.249}$$

and set the initial value of $\hat{g}$ to 1, and the bound of the tracking error, M_ϵ, to 15.

In this situation, the unknown function, $g(x)$, is estimated by the adaptive law eq.(9.36).

$$\dot{\hat{g}} = \gamma \epsilon u_c + S_g \beta_g \gamma \hat{g} \ . \tag{9.250}$$

where $\gamma \triangleq 0.1$. Let $M_g \triangleq 10$. and $\varepsilon_g \triangleq 0.1$. The controller output, u, the controlled trajectory of system output, x, and the tracking error, ϵ, are shown in fig.9.4(a-c), respectively, where the adaptation process stops at $t = 20$ sec. The estimations for the system state, x, and the unknown function, g, are shown in fig.9.5(a-b), respectively.

It is seen that the controlled trajectory converges very well to the desired trajectory and follows this desired trajectory after the adaptation process stops. The training process for the DFLS identifier also converges very well, and the trained identifier predicts the system state satisfactorily after the training stops.

It is observed in fig.9.5(b) that $\hat{g}$ does not converge to the true value of g. This is not surprising because the true value of g is not known, and the adaptive algorithm adjusts $\hat{g}$ in such a way as to reduce the tracking error, ϵ, instead of the estimation error, $(\hat{g} - g)$, and at the same time restrains the estimated value, $\hat{g}$, from exceeding its desired bounds.

9.9.1.C Function $g(x)$ Unknown and Estimated with FLS

Consider the situation where $g(x)$ is an unknown function, and we use a FLS to estimate its value, i.e.,

$$\hat{g} \triangleq \mathbf{\Psi}^T(x')\overline{\mathbf{G}} \ . \tag{9.251}$$

where $\hat{g}$ represents the identifier's output and is the estimate of g. The FLS has one input, x'_1, and one output, $\hat{g}$. Five primary fuzzy sets, B_j, $j = 1,\ldots,5$, are defined in the input universe of discourse of the FLS. Gaussian type membership

functions are used for all the primary fuzzy sets, which are the same as those given in eqs.(9.236-9.238). The adaptive law of this FLS is given in eq.(9.43),

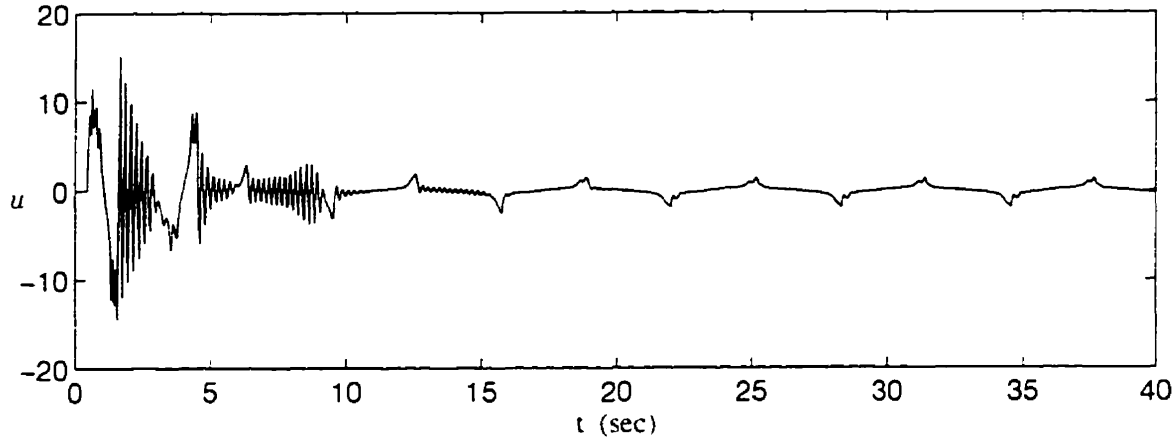
$$\dot{\overline{\mathbf{G}}} = \gamma \epsilon u_c \mathbf{\Psi}(x_1') - S_g \beta_g \gamma \overline{\mathbf{G}} \ . \tag{9.252}$$

where $\gamma \triangleq 0.1$. Let $M_g \triangleq 100$, $\varepsilon_g \triangleq 0.1$, $M_{\overline{\mathbf{G}}} \triangleq \frac{1}{\sqrt{5}} \cdot M_g$, and set all the elements of the parameter vector, $\overline{\mathbf{G}}(0)$, at 1. Define the prefilters as

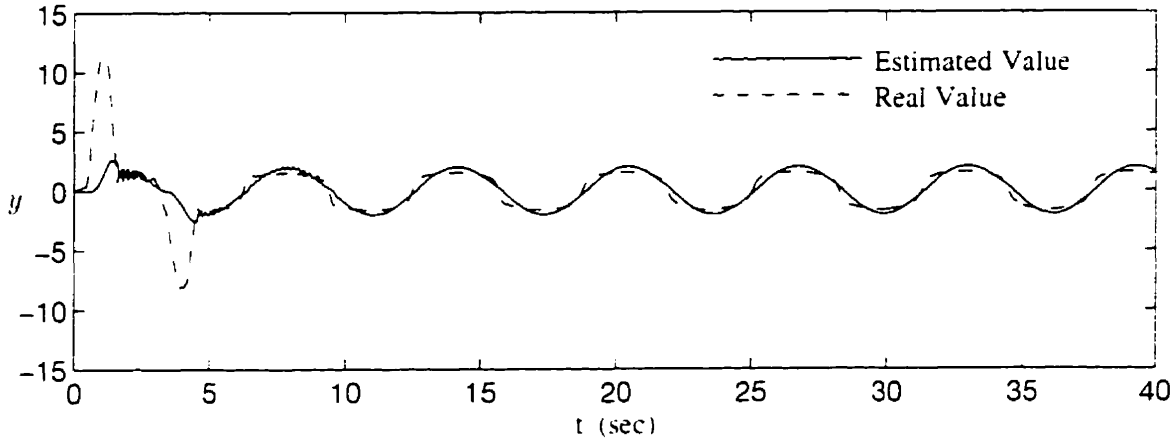
$$\begin{cases} x_1' &= f_1(x_1) \triangleq x_1 \cdot \frac{1}{2} \ , \\ x_2' &= f_2(x_2) \triangleq x_2 \cdot \frac{1}{25} \ . \end{cases} \tag{9.253}$$

The controller output, u, the controlled trajectory of system output, x, and the tracking error, ϵ, are shown in fig.9.6(a-c), respectively, where the adaptation process stops at $t = 20$ sec. The estimations for the system state, x, and the unknown function, g, are shown in fig.9.7(a-b), respectively.

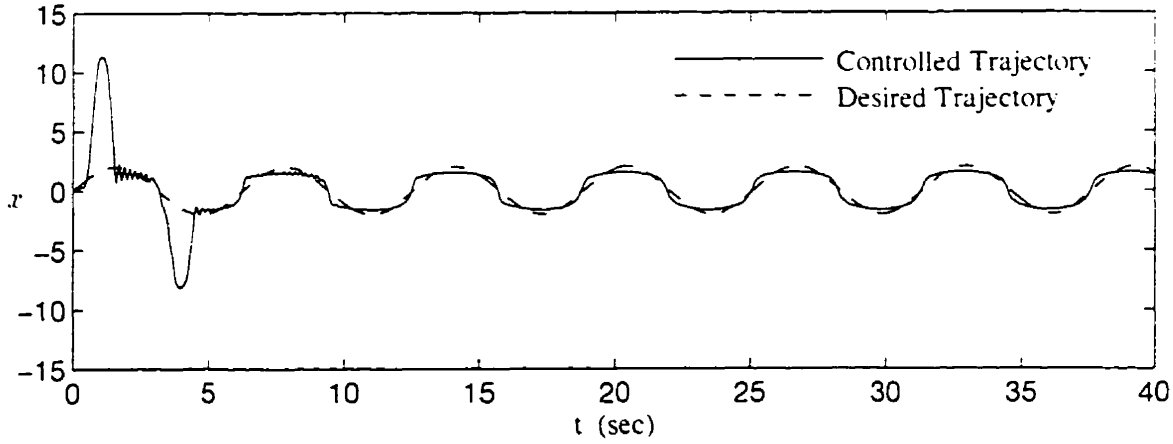
Again, the system performance is very satisfactory. Comparing figs.9.6-9.7 of this system with figs.9.4-9.5 where the unknown function being estimated with self-tuning scheme, it is seen that their performances are generally comparable.



(a) Controller Output



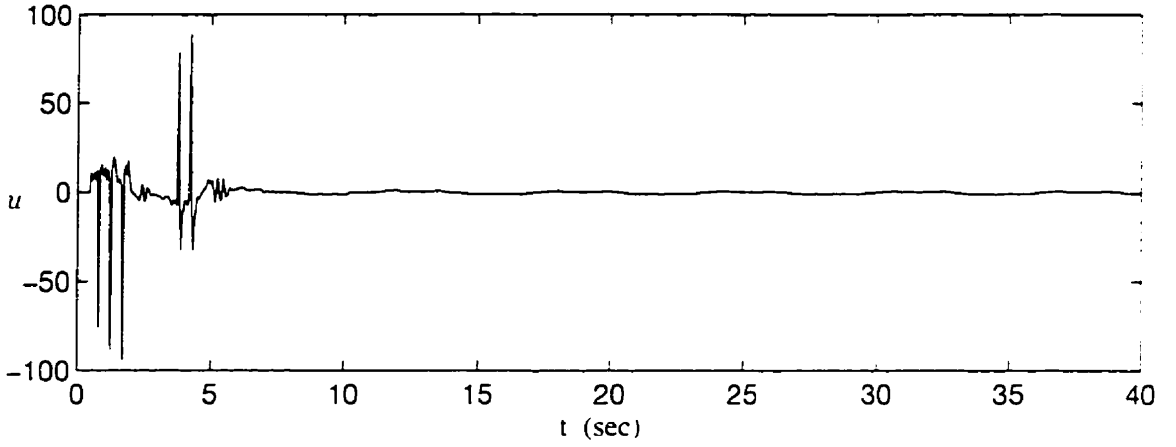
(b) Identifier Output - y



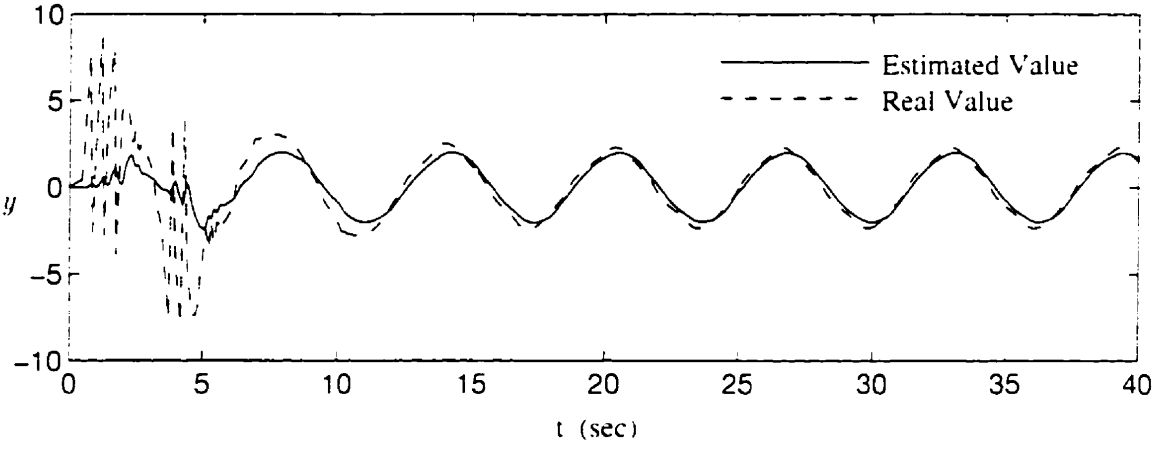
(c) Controlled System Trajectory - x

(Adaptation Terminates at $t = 20$ sec)

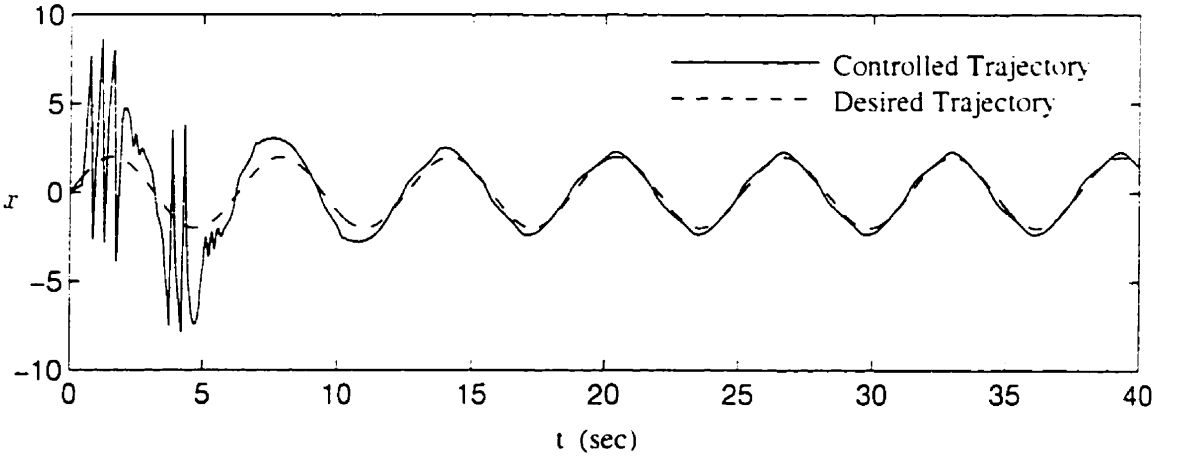
Figure 9.1: Trajectory Control for $M_c \triangleq 15$ -- Example 1 (Function g Known)



(a) Controller Output



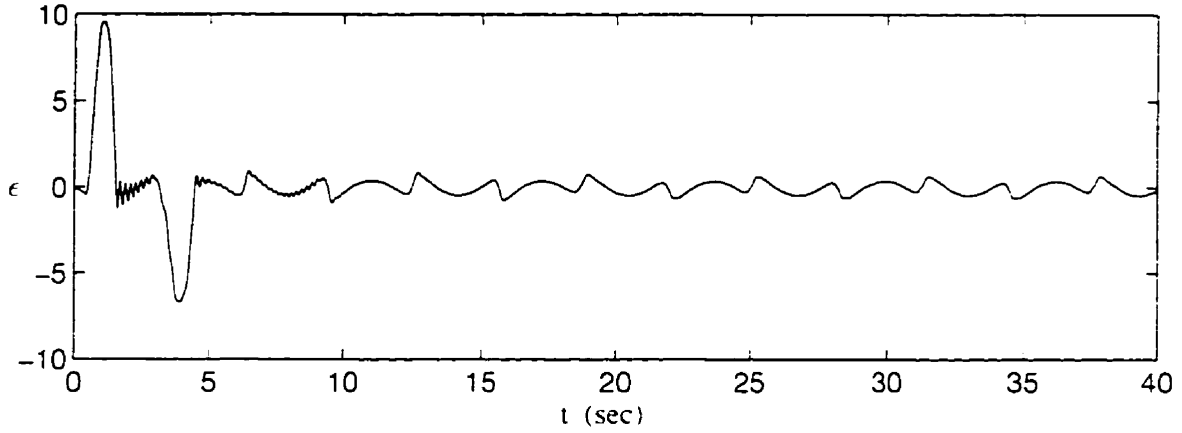
(b) Identifier Output - y



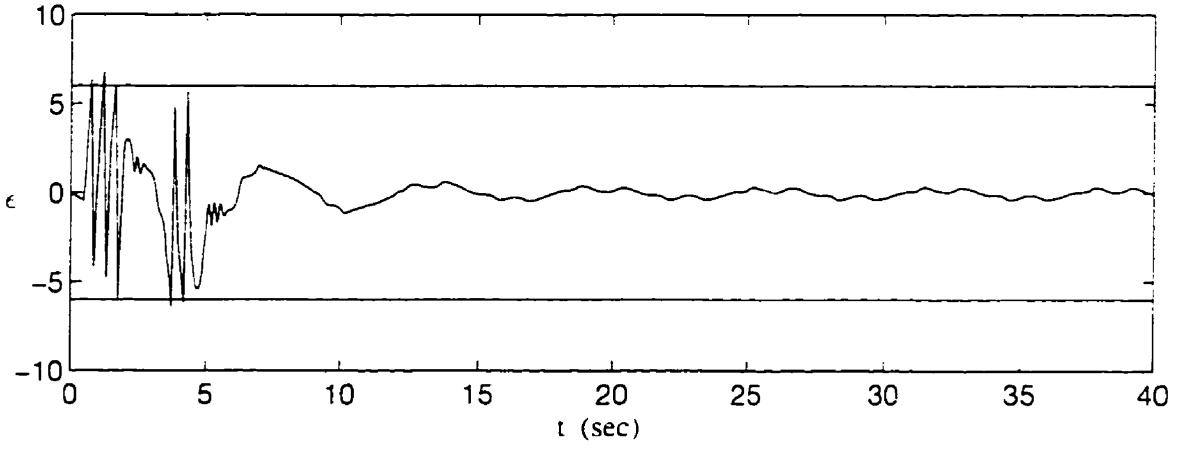
(c) Controlled System Trajectory - x

(Adaptation Terminates at $t = 20$ sec)

Figure 9.2: Trajectory Control for $M_c \triangleq 6$ — Example 1 (Function g Known)

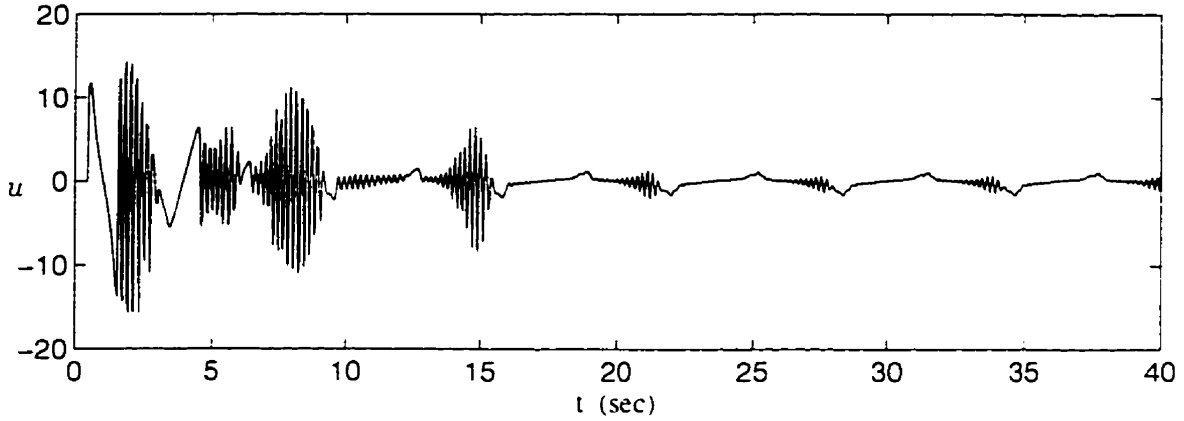


(a) Tracking Error for $M_e \triangleq 15$

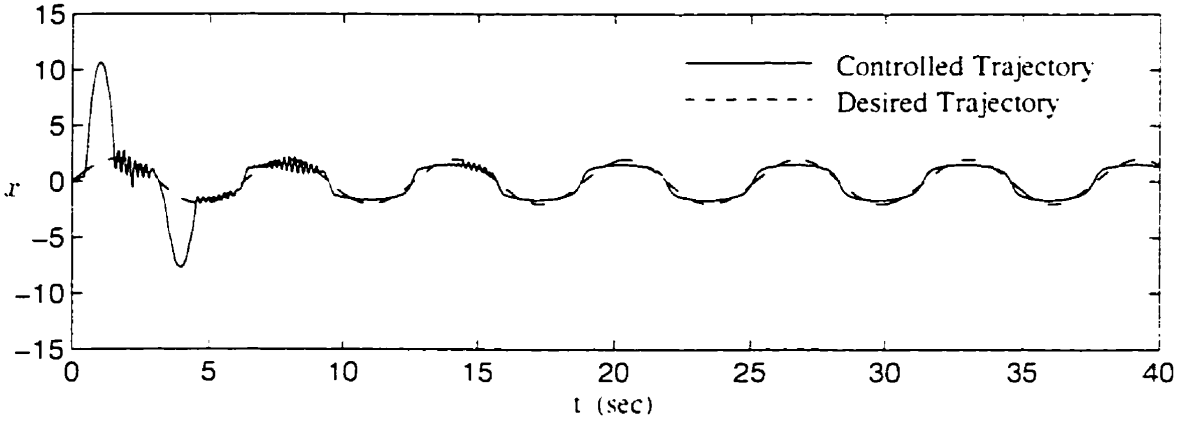


(b) Tracking Error for $M_e \triangleq 6$

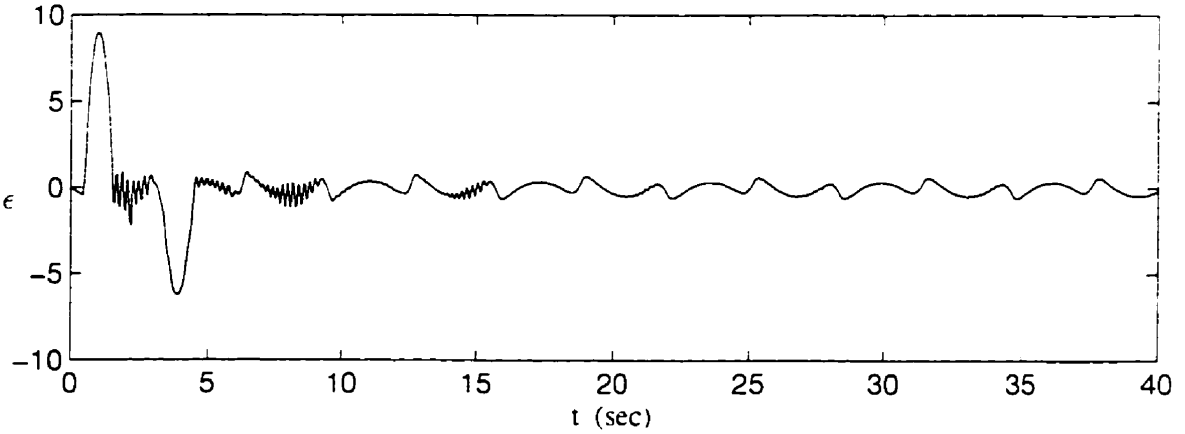
Figure 9.3: Trajectory Errors for Different Error Bounds — Example 1



(a) Controller Output



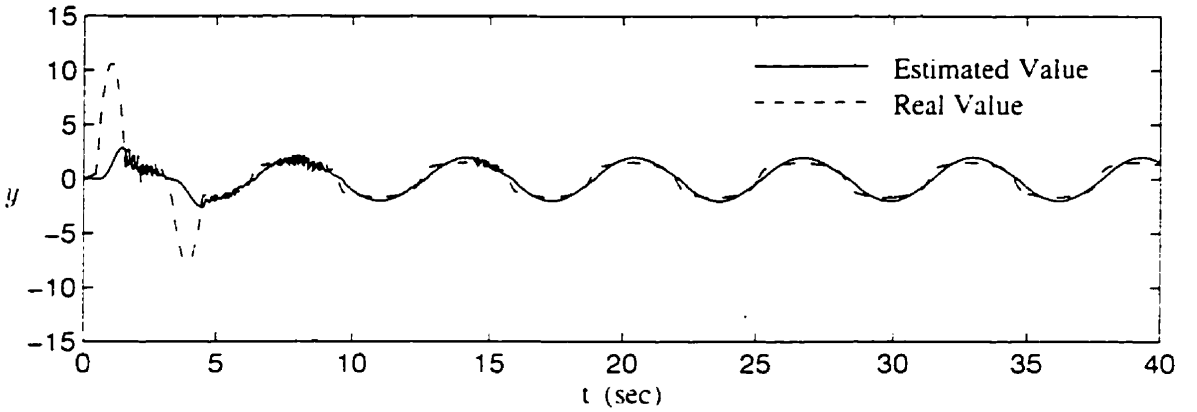
(b) Controlled Trajectory - x



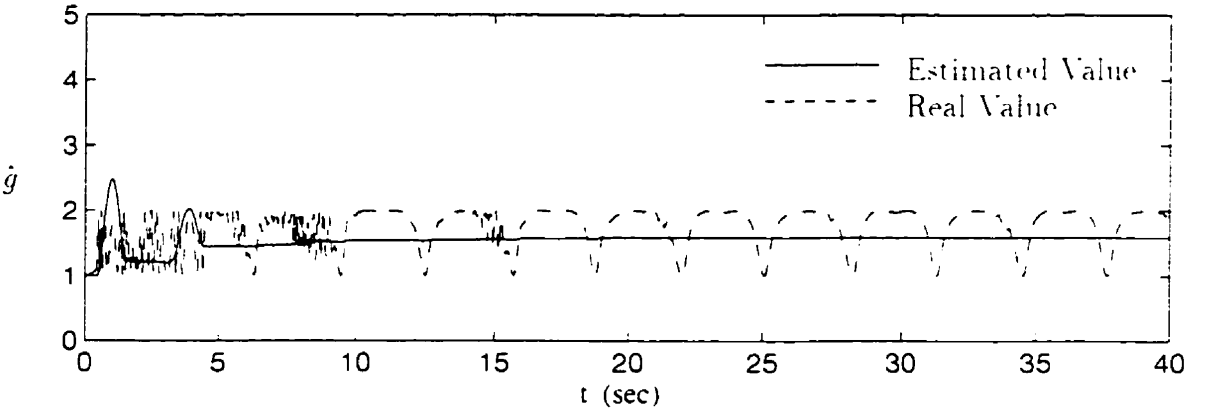
(c) Tracking Error - ϵ

(Adaptation Terminates at $t = 20$ sec)

Figure 9.4: Trajectory Control — Example 1 (Function g Unknown and Estimated with Self-tuning Scheme)

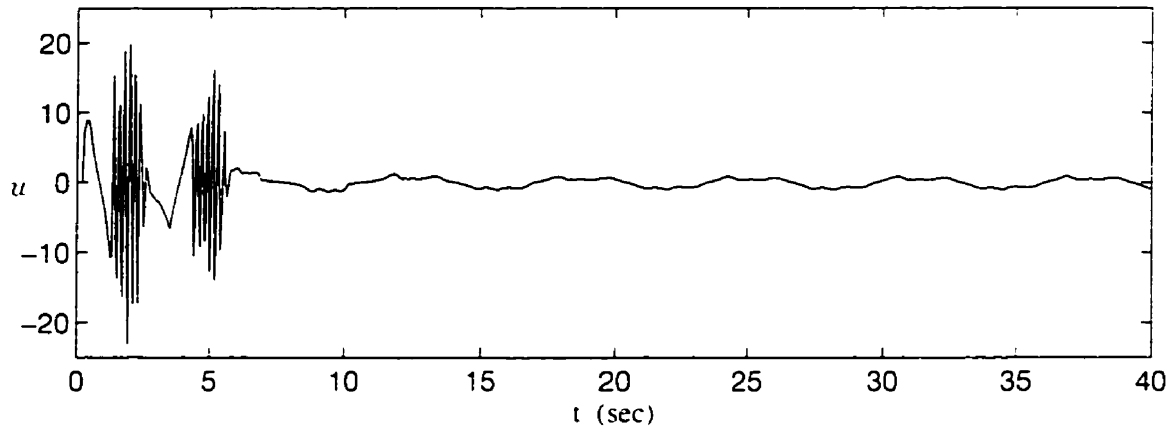


(a) Estimation of State x

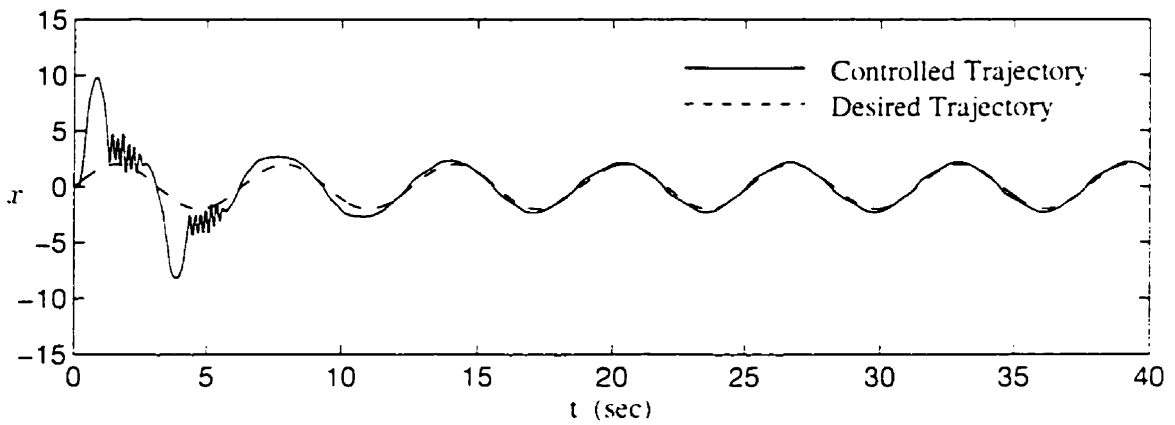


(b) Estimation of Function $g(x)$

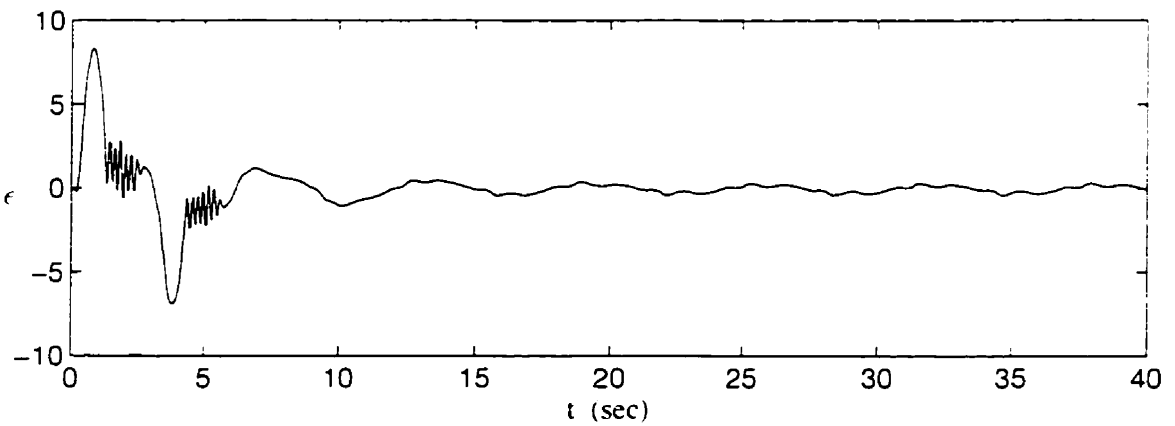
Figure 9.5: Estimation of x and $g(x)$ — Example 1 (Function g Unknown and Estimated with Self-tuning Scheme)



(a) Controller Output



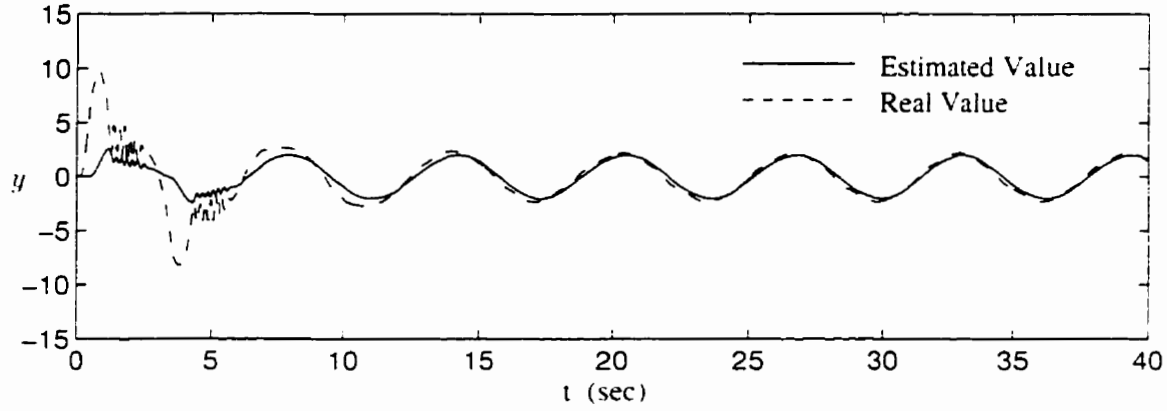
(b) Controlled Trajectory x



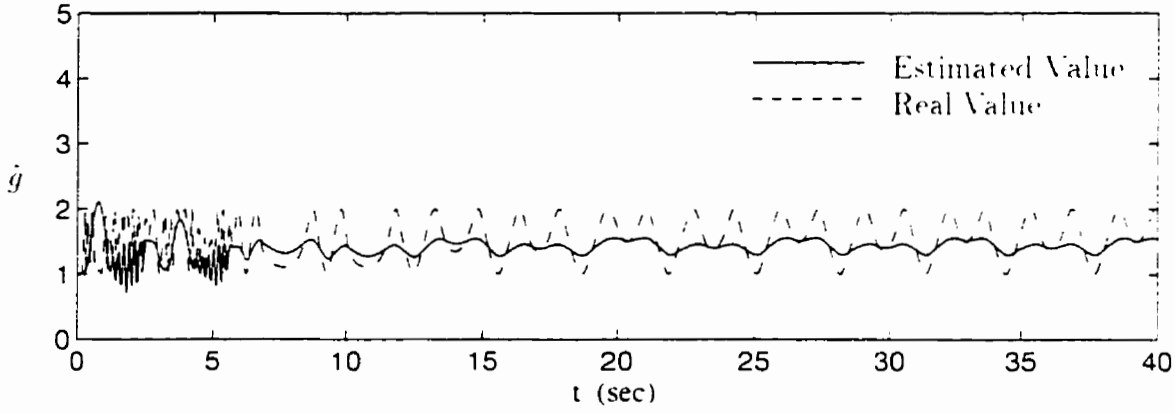
(c) Tracking Error - e

(Adaptation Terminates at $t = 20$ sec)

Figure 9.6: Trajectory Control — Example 1 (Function g Unknown and Estimated with FLS)



(a) Estimation of State x



(b) Estimation of Function $g(x)$

Figure 9.7: Estimation of x and $g(x)$ — Example 1 (Function g Unknown and Estimated with FLS)

9.9.2 Example 2 – Control of a Nonlinear System with Chaotic Behavior

Consider a nonlinear system

$$\begin{cases} \dot{x}_1 &= x_2 \ . \\ \dot{x}_2 &= -x_1^5 - 0.1x_2 + 6\sin t - g(\mathbf{x})u \ . \end{cases} \tag{9.254}$$

which is similar to the system in section 8.4.1, i.e., eq.(8.70), except that an extra control input, u, is present. Our objective here is to command the output of this system, x_1, to track the desired trajectory, $q(t)$, by using the DFLS adaptive control scheme developed in previous sections, where

$$q(t) \triangleq 2\sin(t) \ . \tag{9.255}$$

A DFLS identifier, labeled D, is used to identify the state variable, x_2, i.e.,

$$y \triangleq D(\mathbf{x}, u) \ , \tag{9.256}$$

where $\mathbf{x} \triangleq \{x_1, x_2\}^T$, y represents the identifier's output and is the estimate of x_2. The tracking error, ϵ, and identification error, ξ, are

$$\begin{cases} \epsilon &\triangleq x_1 - q \ . \\ \xi &\triangleq y - x_2 \ . \end{cases} \tag{9.257}$$

The identifier, D, has three inputs, $x_1 \in \mathcal{X}_1$, $x_2 \in \mathcal{X}_2$, $u \in \mathcal{U}$, and one output, $y \in \mathcal{Y}$, where $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{U}$, $\mathcal{Y} \subset R$, are universes of discourse of linguistic variables, x_1, x_2, u and y, respectively. In each of $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{U}$, five primary fuzzy sets, A_{1j_1}, A_{2j_2}, and A_{3j_3}, $j_1, j_2, j_3 = 1, \ldots, 5$, are defined. For conciseness of expression, let $x_3 \triangleq u$, and $\mathbf{z} \triangleq \{x_1, x_2, u\}^T \triangleq \{x_1, x_2, x_3\}^T$. Gaussian type membership functions are used for all the primary fuzzy sets, which are the same as those defined in eqs.(9.236–9.238), for $p = 1, 2, 3$, and $j_p = 1, \ldots, 5$.

For $p = 1.2.3.$ the filtered values of the inputs. x'_p. are defined as

$$\begin{cases} x'_1 &= f_1(x_1) \triangleq x_1 . \\ x'_2 &= f_2(x_2) \triangleq x_2 . \\ x'_3 &= f_3(x_3) \triangleq x_3 \cdot \frac{1}{10} . \end{cases} \tag{9.258}$$

The output of $D.$ $y.$ is defined with the DFLS of eq.(9.4).

$$D(\mathbf{z}'): \quad \dot{y} = -\alpha y + \mathbf{\Theta}^T(\mathbf{z}')\overline{\mathbf{Y}} . \tag{9.259}$$

In this equation.

$$\begin{cases} \mathbf{\Theta}(\mathbf{z}') \triangleq \{\theta_1(\mathbf{z}'). \theta_2(\mathbf{z}'). \ldots. \theta_{125}(\mathbf{z}')\}^T . \\ \overline{\mathbf{Y}} \triangleq \{\overline{y}_1. \overline{y}_2. \ldots. \overline{y}_{125}\}^T . \end{cases} \tag{9.260}$$

and

$$\theta_i(\mathbf{z}') = \frac{\prod_{p=1}^{3} \mu_{A_p^i}(x'_p)}{\sum_{i=1}^{125} \prod_{p=1}^{3} \mu_{A_p^i}(x'_p)} . \qquad i = 1. \ldots. 125. \tag{9.261}$$

where

$$\begin{cases} A_1^i \in \{A_{1j_1}: j_1 = 1. \ldots. 5\} . \\ A_2^i \in \{A_{2j_2}: j_2 = 1. \ldots. 5\} . \\ A_3^i \in \{A_{3j_3}: j_3 = 1. \ldots. 5\} . \end{cases} \tag{9.262}$$

$\overline{\mathbf{Y}}$ is the free parameter vector to be adaptively adjusted with the training law. eq.(9.32).

$$\dot{\overline{\mathbf{Y}}} = -\mathbf{H}[\mathbf{\Theta}(\mathbf{z}')h\xi - \mathbf{\Theta}(\mathbf{x}'.0)(\mathbf{B}^T\mathbf{Pe})] - S\beta\mathbf{H}\overline{\mathbf{Y}} . \tag{9.263}$$

Using a numerical difference to approximate the derivative yields

$$\begin{aligned} \overline{\mathbf{Y}}(kT + T) &= \overline{\mathbf{Y}}(kT) - T\mathbf{H}[\ \mathbf{\Theta}(\mathbf{z}'(kT))h\xi(kT) \\ &\quad - \mathbf{\Theta}(\mathbf{x}'(kT).0)(\mathbf{B}^T\mathbf{Pe}(kT)) + S\beta\overline{\mathbf{Y}}(kT)\ ] . \end{aligned} \tag{9.264}$$

where T is the time incremental step.

Let $\alpha = \alpha_1 \triangleq 10$. We set the time increment, T, to 0.1 sec, the bound of the parameter vector, $M_{\overline{\mathbf{Y}}}$, to 10^3, the bound of the function f, to $M_f = |x_1|^5 + 10$, and the bound of the tracking error, $M_{\mathbf{e}}$, to 20. Let the initial values of the elements of the parameter vector, $\overline{\mathbf{Y}}(0)$, be random numbers uniformly distributed in $(-1, 1)$, and assume the system to be controlled is initially at rest, i.e., $\mathbf{x}(0) = \{0.0\}^T$. Now, we consider the following situations.

9.9.2.A Function $g(\mathbf{x})$ Known

Consider the situation where $g(\mathbf{x})$ is a known function, say,

$$g \equiv 1 \ . \tag{9.265}$$

In this situation, we require no identifier to estimate g. Let $\mathbf{Q} \triangleq \mathbf{I}$, where $\mathbf{I}$ is an identity matrix, and let the gain matrix, $\mathbf{H}$, be a diagonal matrix with all diagonal elements equal to 1.5, and set the weight parameter, h, to 15. The controller output, u, the identifier output, y, and the controlled trajectory of system output, x_1, are shown in fig.9.8(a-c), respectively, where the adaptation process stops at $t = 20$ sec. The controlled trajectory in phase plane, $x_1 - x_2$, is shown in fig.9.9.

It is seen that the identification process converges quite well, and the control strategy is quite effective, considering the lack of knowledge of the system dynamics and the limited training effort.

9.9.2.B Function $g(\mathbf{x})$ Unknown and Estimated with Self-Tuning Scheme

Consider the situation where $g(\mathbf{x})$ is an unknown function,

$$g = \frac{2}{1 + \cos^2(x_1)} \ , \tag{9.266}$$

which will be estimated with the self-tuning algorithm. The adaptive law for adjusting $\hat{g}$ is given by eq.(9.36),

$$\dot{\hat{g}} = \gamma(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c + S_g\beta_g\gamma\hat{g} \ . \tag{9.267}$$

Let $\mathbf{Q} \triangleq 10 \cdot \mathbf{I}$, where $\mathbf{I}$ is an identity matrix. Set the initial value of $\hat{g}$ to 1, the weight parameter, h, to 20, and the gain matrix, $\mathbf{H}$, is defined as a diagonal matrix with diagonal elements equal to 2. Let $\gamma \triangleq 0.01$, $M_g \triangleq 10$, and $\varepsilon_g \triangleq 0.1$. The controller output, u, the controlled trajectory of system output, x_1, and the controlled trajectory in phase plane, $x_1 - x_2$, are shown in fig.9.10(a-c), respectively, where the adaptation process stops at $t = 30$ sec. The estimations for the state variable, x_2, and the function, $g(\mathbf{x})$, are shown in fig.9.11(a-b), respectively.

Again, the controlled trajectory converges very well to the desired trajectory and follows the desired trajectory after the adaptation process stops. The training process for the DFLS identifier also converges quite well, and the trained identifier predicts the system state satisfactorily after the training ends.

9.9.2.C $g(x)$ Estimated with FLS

Here we use a FLS identifier to estimate the unknown function, $g(\mathbf{x})$, of eq.(9.266),

$$\hat{g} \triangleq \mathbf{\Psi}^T(\mathbf{x}')\overline{\mathbf{G}} \ . \tag{9.268}$$

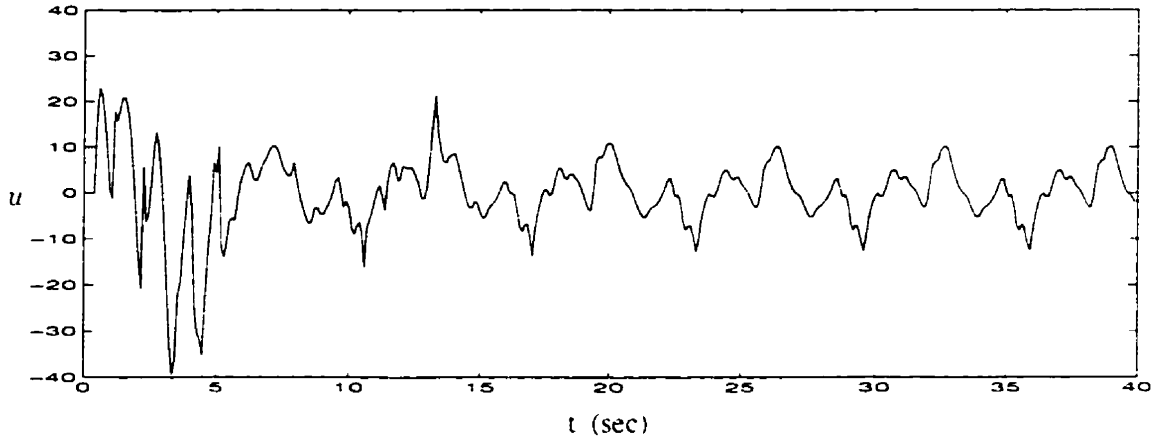
where $\hat{g}$ represents the identifier's output and is the estimate of g. The FLS has two inputs, x_1', x_2', and one output, $\hat{g}$. Five primary fuzzy sets, B_{j_p}, $p = 1, 2$, $j_1, j_2 = 1, \ldots, 5$, are defined in both universes of discourse in the input space of the FLS. Gaussian type membership functions are used for all the primary fuzzy sets, which are the same as those given in eqs.(9.236-9.238). The adaptive law for this FLS is given in eq.(9.43),

$$\dot{\overline{\mathbf{G}}} = \gamma \mathbf{\Psi}(\mathbf{x}')(\mathbf{B}^T\mathbf{P}\mathbf{e})u_c - S_g \beta_g \gamma \overline{\mathbf{G}} \ . \tag{9.269}$$

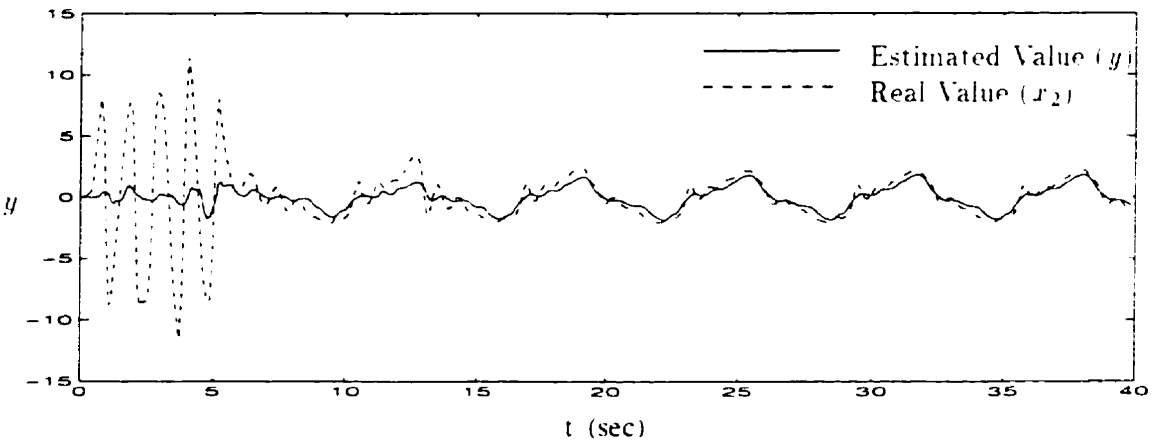
Let $\gamma \triangleq 0.001$, $M_g \triangleq 100$, $\varepsilon_g \triangleq 0.1$, $M_{\overline{\mathbf{G}}} \triangleq \frac{1}{5} M_g$, $\mathbf{Q} \triangleq 10 \cdot \mathbf{I}$, and set the initial value of all the elements of the parameter vector, $\overline{\mathbf{G}}(0)$, to 1. The gain matrix, $\mathbf{H}$, is defined as a diagonal matrix with all diagonal elements 1, and the weight parameter, h, is 20.

The controller output, u, the controlled trajectory of system output, x_1, and the controlled trajectory in phase plane, $x_1 - x_2$, are shown in fig.9.12(a-c), respectively, where the adaptation process stops at $t = 30$ sec. The estimations for the state variable, x_2, and the function, $g(\mathbf{x})$, are shown in fig.9.13, respectively.

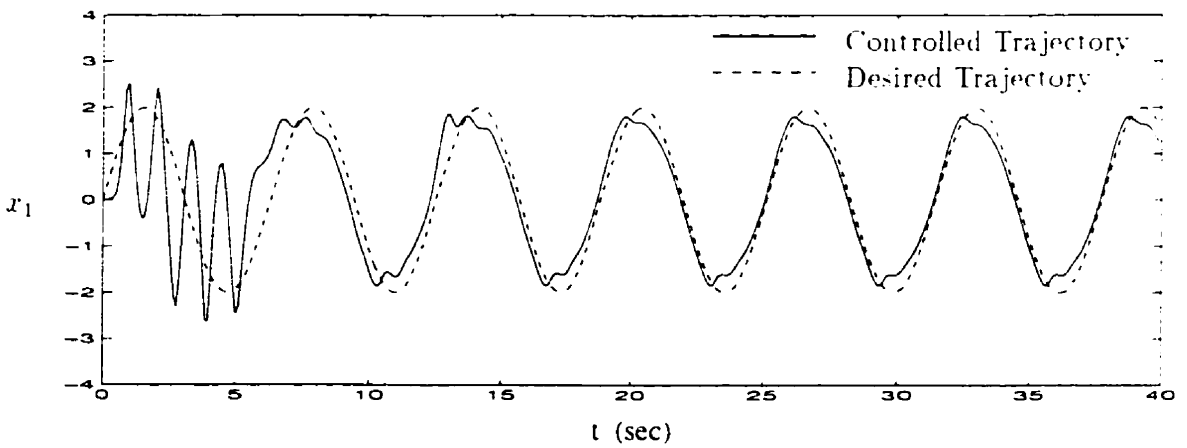
The controlled trajectory converges very well to the desired trajectory and follows the desired trajectory after the adaptation process stops. The training process for DFLS identifier also converges quite well, and the trained identifier predicts system state satisfactorily after the training stops. The performance of this system is comparable to that where g is estimated with the self-tuning scheme.



(a) Controller Output - u



(b) Identifier Output - y



(c) Controlled State Trajectory - x_1

(Adaptation Terminates at $t = 20$ sec)

Figure 9.8: Trajectory Control — Example 2 (Function g Known)

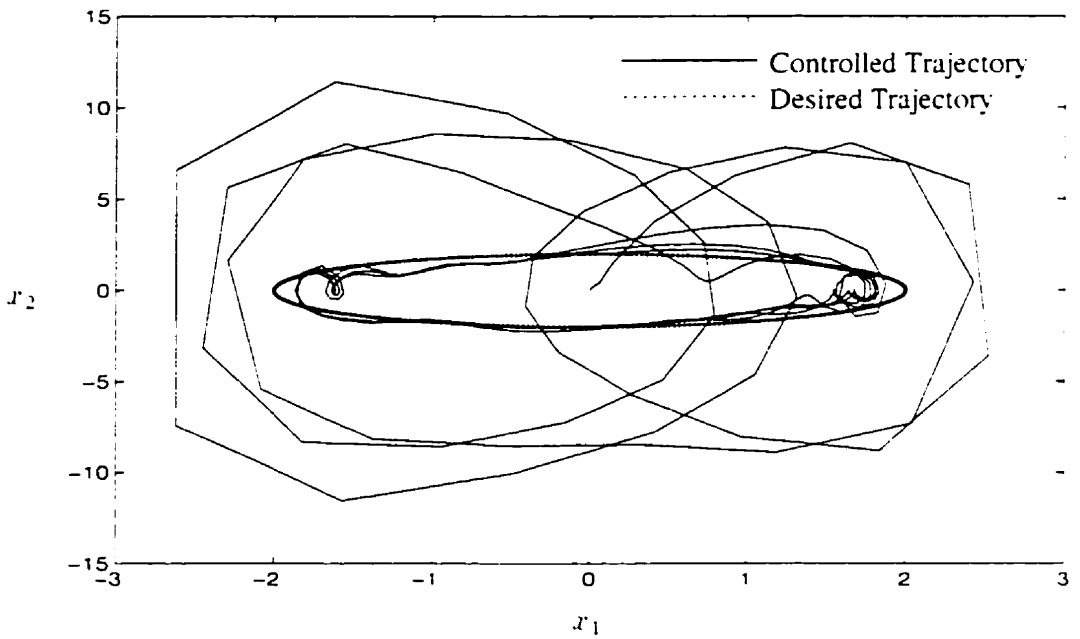
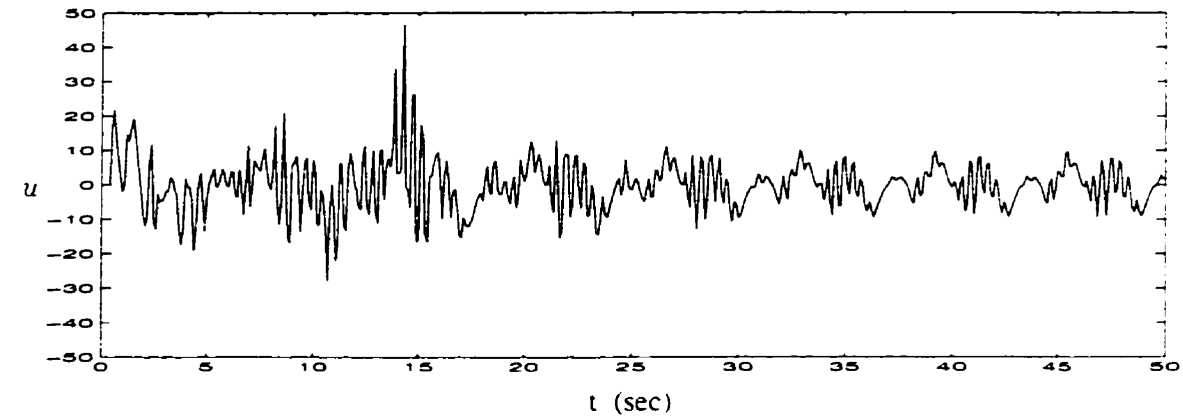
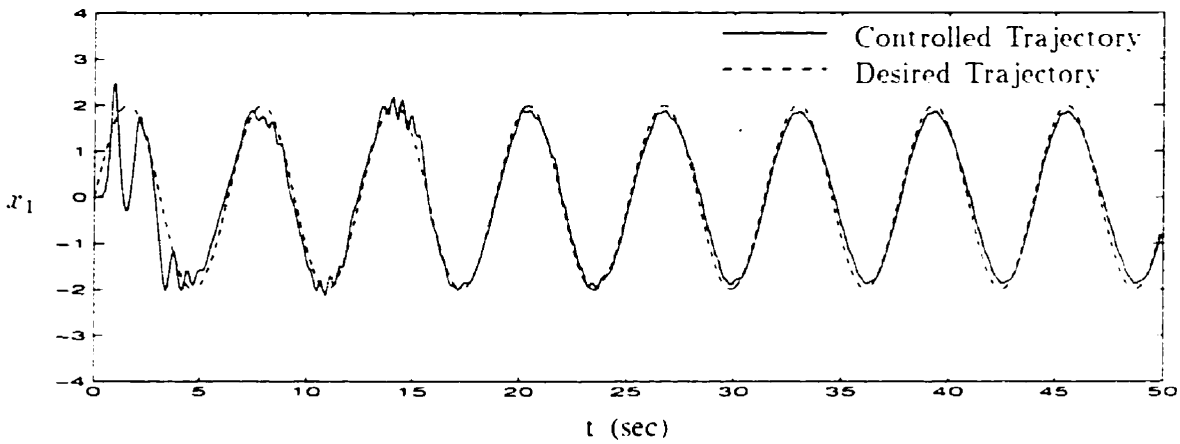


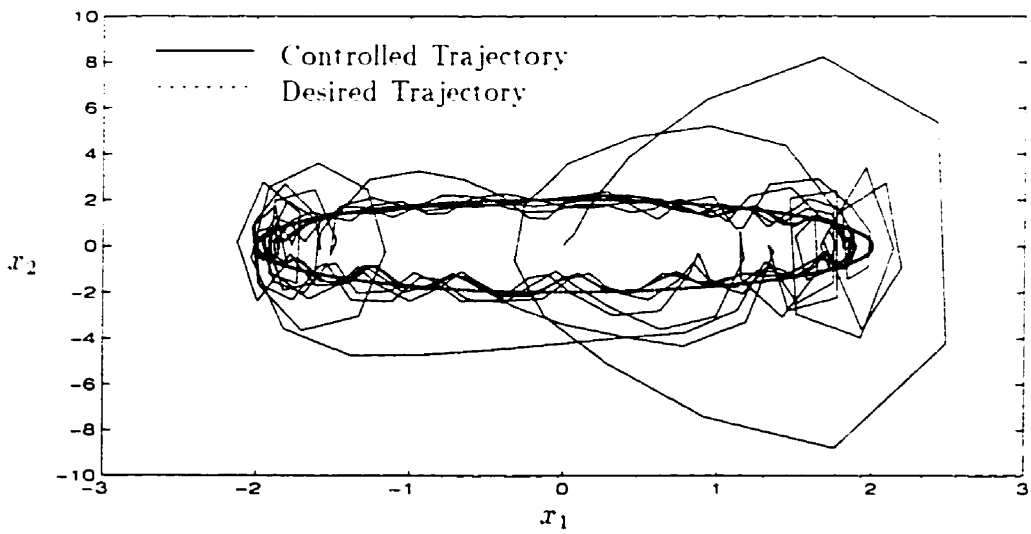
Figure 9.9: Phase Plane Expression of Controlled Trajectory — Example 2 (Function g Known)



(a) Controller Output - u



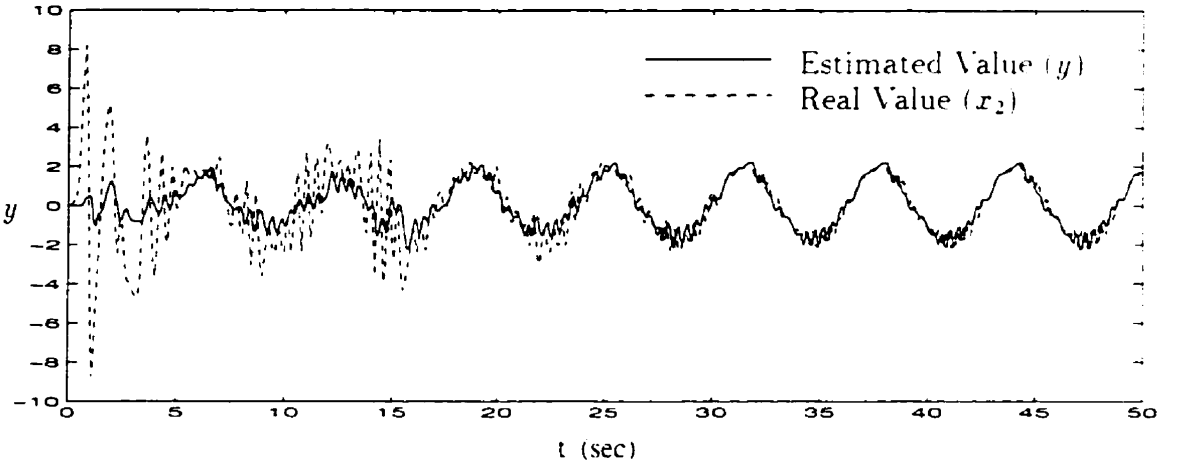
(b) Controlled Trajectory - x_1



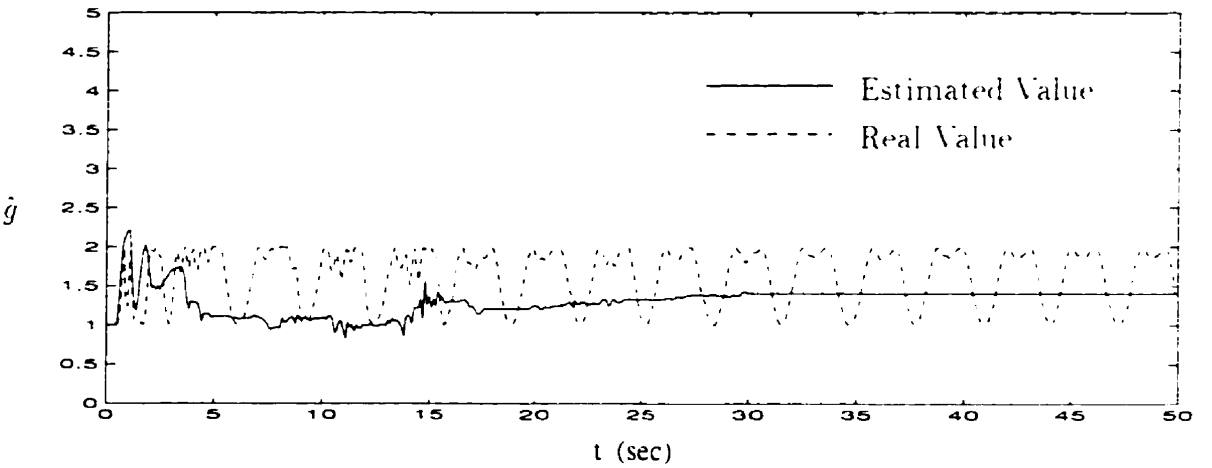
(c) Phase Plane Expression of Controlled Trajectory

(Adaptation Terminates at $t = 30$ sec)

Figure 9.10: Trajectory Control — Example 2 (Function g Unknown and Estimated with Self-tuning Scheme)

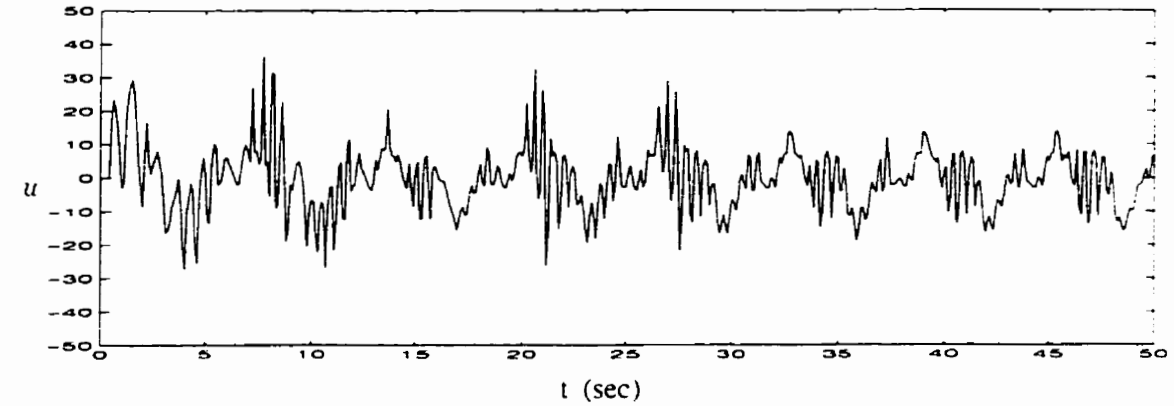


(a) Estimation of State x_2

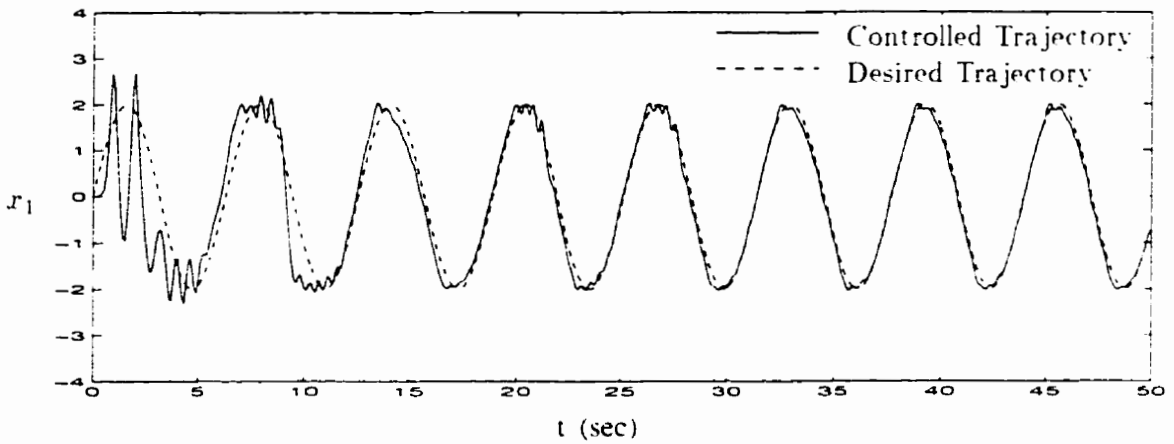


(b) Estimation of Function $g(\mathbf{x})$

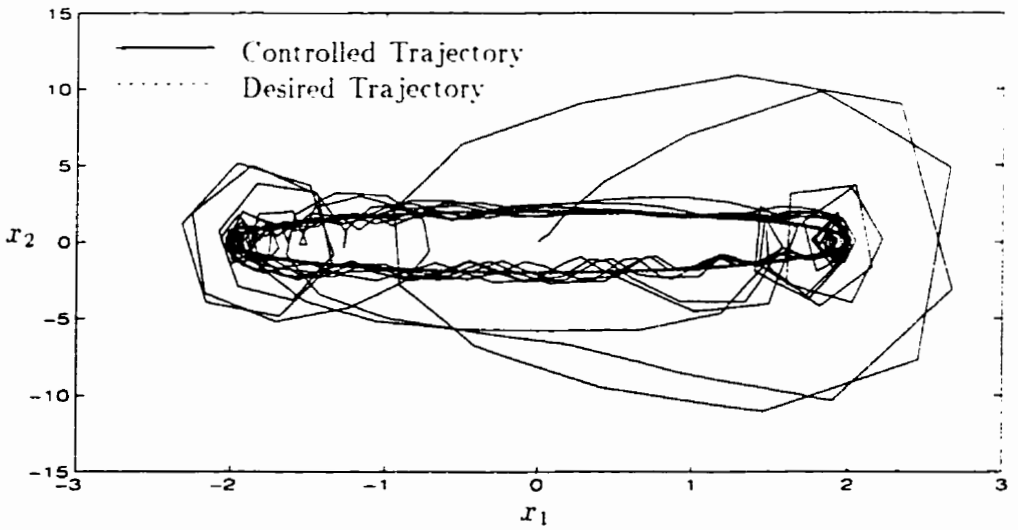
Figure 9.11: Estimation of x_2 and $g(\mathbf{x})$ — Example 2 (Function g Unknown and Estimated with Self-tuning Scheme)



(a) Controller Output - u



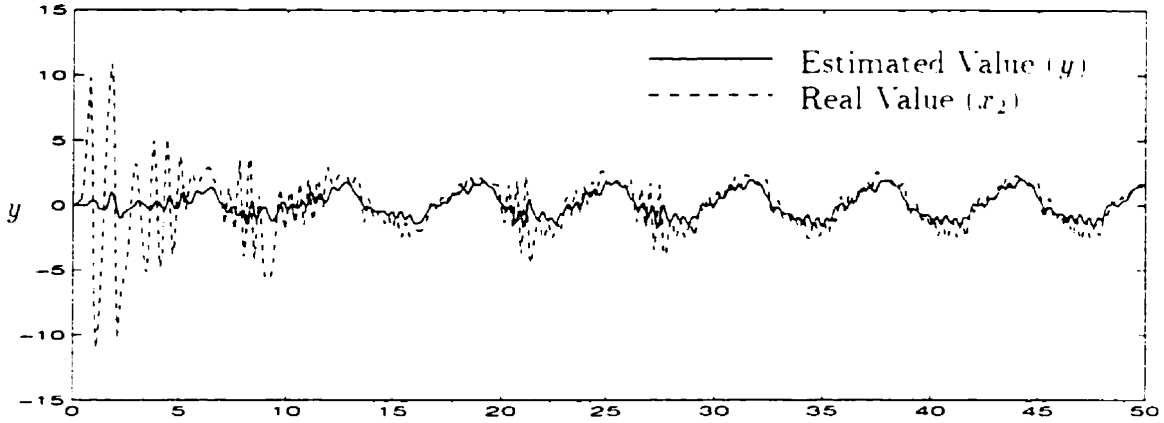
(b) Controlled Trajectory - x_1



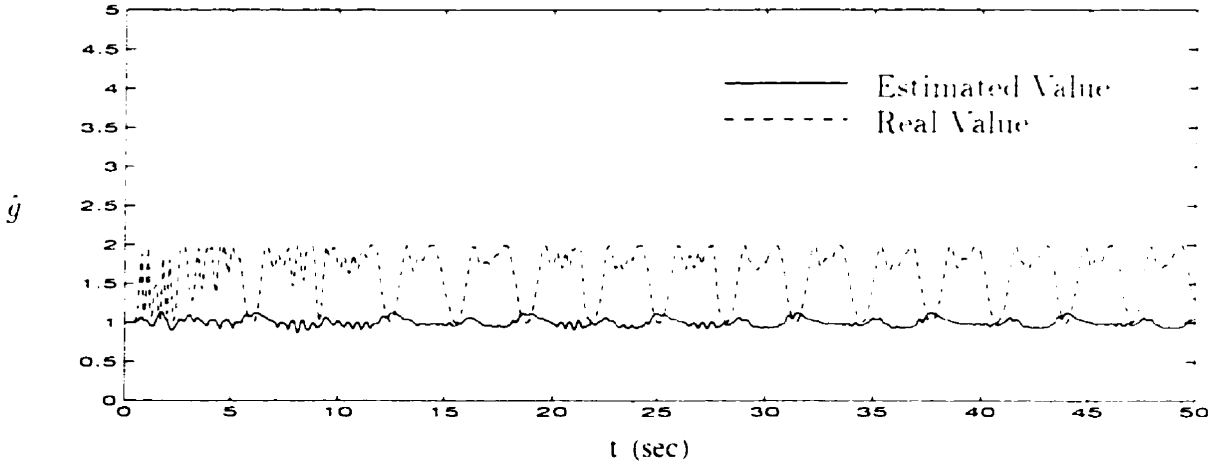
(c) Phase Plane Expression of Controlled Trajectory

(Adaptation Terminates at $t = 30$ sec)

Figure 9.12: Trajectory Control — Example 2 (Function g Unknown and Estimated with FLS)



(a) Estimation for State x_2



(b) Estimation of Function $g(\mathbf{x})$

Figure 9.13: Estimation of x_2 and $g(\mathbf{x})$ — Example 2 (Function g Unknown and Estimated with FLS)

9.10 Experimental Demonstration – Trajectory Control of a Mechanical Manipulator

Consider the single link manipulator described in Appendix B, which is characterized by a very flexible link, significant nonlinear joint friction, and an unknown internal motor speed regulation loop. Our objective here is to command the tip of the link to follow a desired trajectory, $q(t)$, by using the DFLS adaptive control algorithm.

The manipulator is schematically shown in fig.9.14, where $L = 0.54$ m is the length of the beam, $v(L,t)$ is the vibrational displacement of the tip of the beam, $\Phi(t)$ is the angular displacement of the rigid body motion of the beam, and $w(t)$ is denoted as the displacement of the tip of the beam, i.e.,

$$w(t) \triangleq L\Phi(t) + v(L,t) . \tag{9.270}$$

In the experimental setup, the variables, $\Phi(t)$, $\dot{\Phi}(t)$, and $v(L,t)$, can be directly mea-

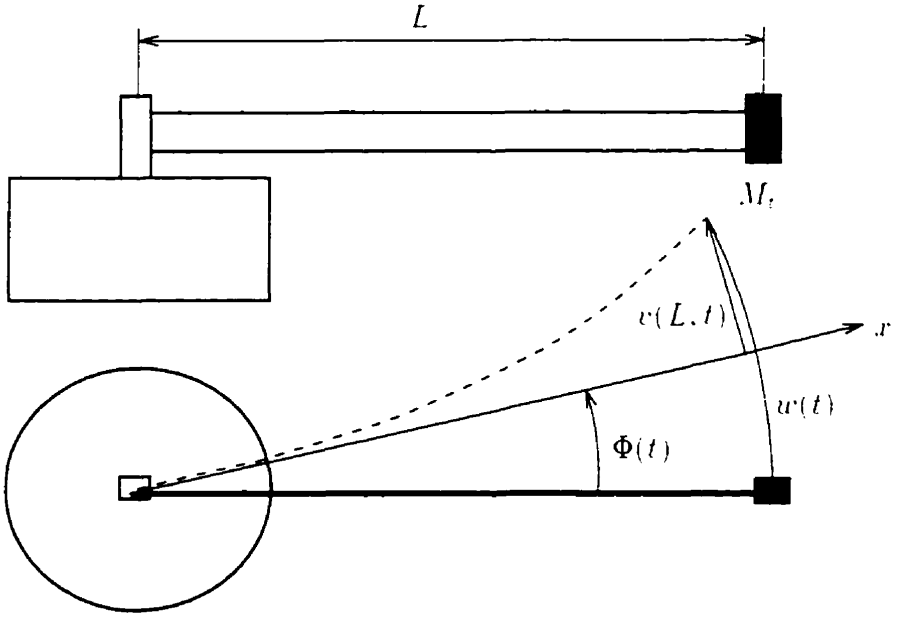


Figure 9.14: Illustration of a Single Link Manipulator

sured with sensors, while $\dot{v}(L,t)$ is obtained by numerical differentiation of $v(L,t)$.

i.e.,

$$\dot{v}(L,t) \triangleq \frac{v(L,t) - v(L,t-T)}{T} . \tag{9.271}$$

As this operation may amplify noise significantly, a low pass filter is used to filter out high frequency noises. A second order low pass digital Butterworth filter (in MATLAB) is used here.

We assume that the system is of the following type

$$\ddot{w} = f(w,\dot{w}) + g(w,\dot{w})V_m , \tag{9.272}$$

where V_m is the command voltage of the motor, f and g are unknown continuous functions. Let

$$\begin{cases} w_1 \triangleq w , \\ w_2 \triangleq \dot{w} , \end{cases} \tag{9.273}$$

then

$$\begin{cases} \dot{w}_1 = w_2 , \\ \dot{w}_2 = f(\mathbf{w}) + g(\mathbf{w})V_m , \end{cases} \tag{9.274}$$

where $\mathbf{w} \triangleq \{w_1, w_2\}^T$.

A DFLS identifier, labeled D, is used to identify the state variable, w_2, i.e.,

$$y \triangleq D(\mathbf{w}, V_m) , \tag{9.275}$$

where y represents the identifier's output and is the estimate of w_2. The tracking error, ϵ, and identification error, ξ, are

$$\begin{cases} \epsilon \triangleq w_1 - q , \\ \xi \triangleq y - w_2 , \end{cases} \tag{9.276}$$

For conciseness of expression, let

$$\begin{cases} x_1 \triangleq \Phi \ , \\ x_2 \triangleq \dot{\Phi} \ , \\ x_3 \triangleq w_1 \ , \\ x_4 \triangleq w_2 \ , \\ x_5 \triangleq V_m \ . \end{cases} \tag{9.277}$$

Further considering eq.(9.270) we have

$$y = D(\Phi,\ \dot{\Phi},\ v(L,t)\ ,\dot{v}(L,t)\ ,V_m) = D(\Phi,\ \dot{\Phi},\ w_1\ ,w_2\ ,V_m) \triangleq D(\mathbf{z})\ . \tag{9.278}$$

where $\mathbf{z} \triangleq \{x_1, x_2, x_3, x_4, x_5\}^T$. The identifier, D, has five inputs, $x_p \in \mathcal{X}_p$, $p = 1,\ldots,5$, and one output, $y \in \mathcal{Y}$, where $\mathcal{X}_p$, $\mathcal{Y} \subset R$ are universes of discourse of their respective linguistic variables. In each of $\mathcal{X}_p$, three primary fuzzy sets, A_{pj_p}, $p = 1,\ldots,5$, $j_p = 1,2,3$, are defined. Gaussian type membership functions are used for all the primary fuzzy sets.

$$\begin{aligned} \mu_{A_{p1}}(x'_p) &\triangleq \begin{cases} 1\ , & if \quad x'_p < \overline{x}_{p1}\ , \\ \exp[-\frac{1}{2}(\frac{x'_p-\overline{x}_{p1}}{\sigma_{p1}})^2]\ , & if \quad x'_p \geq \overline{x}_{p1}\ , \end{cases} \\ \mu_{A_{p2}}(x'_p) &\triangleq \exp[-\frac{1}{2}(\frac{x'_p-\overline{x}_{p2}}{\sigma_{p2}})^2]\ , \\ \mu_{A_{p3}}(x'_p) &\triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{x'_p-\overline{x}_{p3}}{\sigma_{p3}})^2]\ , & if \quad x'_p < \overline{x}_{p3}\ , \\ 1\ , & if \quad x'_p \geq \overline{x}_{p3}\ , \end{cases} \end{aligned} \tag{9.279}$$

Let the shape parameters of all the primary fuzzy sets be 0.45, i.e.,

$$\sigma_{pj_p} \triangleq 0.45\ , \tag{9.280}$$

and let the position parameters, $\overline{x}_{pj_p}$, be,

$$\{\overline{x}_{p1},\ \overline{x}_{p2},\ \overline{x}_{p3}\}^T \triangleq \{-1,\ 0,\ 1\}^T\ . \tag{9.281}$$

In eq.(9.279), x'_p represents the filtered value of the DFLS input, x_p, that is, for $p = 1, \ldots, 5$.

$$x'_p \triangleq f_p(x_p) . \tag{9.282}$$

The output of D, y, is defined via the DFLS of eq.(9.4).

$$D(\mathbf{z}'): \quad \dot{y} = -\alpha y + \boldsymbol{\Theta}^T(\mathbf{z}')\overline{\mathbf{Y}} . \tag{9.283}$$

In this equation,

$$\begin{cases} \boldsymbol{\Theta}(\mathbf{z}') \triangleq \{\theta_1(\mathbf{z}'), \theta_2(\mathbf{z}'), \ldots, \theta_{243}(\mathbf{z}')\}^T , \\ \overline{\mathbf{Y}} \triangleq \{\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_{243}\}^T , \end{cases} \tag{9.284}$$

and

$$\theta_i(\mathbf{z}') = \frac{\prod_{p=1}^{5} \mu_{A_p^i}(x'_p)}{\sum_{i=1}^{243} \prod_{p=1}^{5} \mu_{A_p^i}(x'_p)} , \qquad i = 1, \ldots, 243, \tag{9.285}$$

where, for $p = 1, \ldots, 5$, $i = 1, \ldots, 243$,

$$A_p^i \in \{A_{p,j_i} : j_i = 1, 2, 3\} . \tag{9.286}$$

$\overline{\mathbf{Y}}$ is the free parameter vector to be adaptively adjusted with the training law, eq.(9.32).

$$\dot{\overline{\mathbf{Y}}} = -\mathbf{H}[\boldsymbol{\Theta}(\mathbf{z}')h\xi - \boldsymbol{\Theta}(\mathbf{x}', 0)(\mathbf{B}^T\mathbf{P}\mathbf{e})] - S\beta\mathbf{H}\overline{\mathbf{Y}} . \tag{9.287}$$

where $\mathbf{x}' \triangleq \{x'_1, x'_2, x'_3, x'_4\}^T$. Using a numerical difference to approximate the derivative gives

$$\begin{aligned} \overline{\mathbf{Y}}(kT + T) = \overline{\mathbf{Y}}(kT) - T\mathbf{H}[& \boldsymbol{\Theta}(\mathbf{z}'(kT))h\xi(kT) \\ & - \boldsymbol{\Theta}(\mathbf{x}'(kT), 0)(\mathbf{B}^T\mathbf{P}\mathbf{e}(kT)) + S\beta\overline{\mathbf{Y}}(kT)] . \end{aligned} \tag{9.288}$$

where T is the time incremental step.

The unknown function g is estimated with FLS approach.

$$\hat{g} \triangleq \mathbf{\Psi}^T(\mathbf{x}')\overline{\mathbf{G}} \ . \tag{9.289}$$

where $\hat{g}$ represents the identifier's output and is the estimate of g. The FLS has four inputs, x'_1, x'_2, x'_3, x'_4, and one output, $\hat{g}$. Three primary fuzzy sets, B_{pj_p}, $p = 1,2,3,4$, $j_p = 1,2,3$, are defined in all the universes of discourse in the input space of the FLS. Gaussian type membership functions are used for all the primary fuzzy sets, which are the same as those given in eqs.(9.279-9.281). The adaptive law for this FLS is given in eq.(9.43).

$$\dot{\overline{\mathbf{G}}} = \gamma \mathbf{\Psi}(\mathbf{x}')(\mathbf{B}^T\mathbf{P}\mathbf{e})(u_c - S_g\beta_g)\overline{\mathbf{G}} \ . \tag{9.290}$$

The command voltage is restricted in $[-7V, 7V]$, i.e.,

$$-7V \le V_{in} \le 7V \ . \tag{9.291}$$

Correspondingly, the supervisory control term is not required in this control system. Let $\alpha = \alpha_1 \triangleq 10$. Set the time increment, T, to 0.026 sec, the bounds of the parameter vectors, $M_{\overline{\mathbf{Y}}}$ and M_g, to 10^4, and ε_g to 0.01. Let $M_{\overline{\mathbf{G}}} \triangleq \frac{1}{9}M_g$, and set the gain matrix, $\mathbf{H}$, to be a diagonal matrix with all diagonal elements equal to 5, and set the gain, γ, to 5, and the weight parameter, h, to 2. $\mathbf{Q}$ is defined as a diagonal matrix with all diagonal elements set to 4.

The filtered values of inputs, x'_p, $p = 1,\ldots,5$, are defined as

$$\begin{cases} x'_1 = f_1(x_1) \triangleq \Phi(t) \cdot \frac{1}{4} \ , \\ x'_2 = f_2(x_2) \triangleq \dot{\Phi}(t) \cdot \frac{1}{10} \ , \\ x'_3 = f_3(x_3) \triangleq w(t) \cdot \frac{1}{2} \ , \\ x'_4 = f_3(x_4) \triangleq \dot{w}(t) \cdot \frac{1}{5} \ , \\ x'_5 = f_3(x_5) \triangleq V_{in}(t) \cdot \frac{1}{5} \ . \end{cases} \tag{9.292}$$

The initial values of the parameter vectors, $\overline{\mathbf{Y}}(0)$ and $\overline{\mathbf{G}}(0)$, are obtained by simulation of the dynamic model of the manipulator specified in section 8.4.3. That is,

the DFLS controller outlined above is used to control the trajectory of the manipulator represented by the dynamic model in section 8.4.3 to follow various sinusoidal trajectories of different frequencies and magnitudes. The final parameter vectors, $\overline{\mathbf{Y}}$ and $\overline{\mathbf{G}}$, obtained in those simulations are used as initial values in the experiments. The mechanical system is initially at rest, i.e., $\mathbf{w}(0) = \{0.0\}^T$.

For a desired trajectory,

$$q(t) = 0.4\sin(0.8\pi t + \frac{\pi}{2}) - 0.4 \ . \tag{9.293}$$

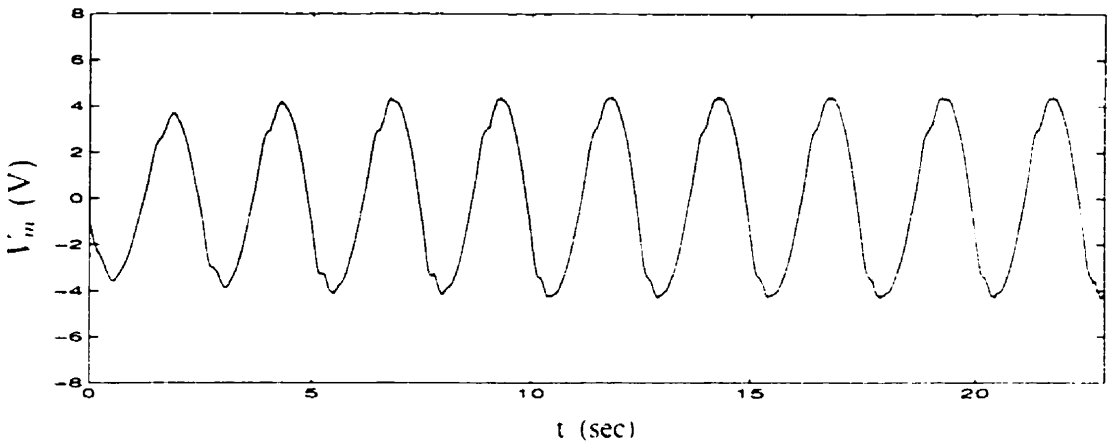
the command voltage, V_m, and the controlled trajectory, w, are shown in fig.9.15(a-b), the estimations for variable $\dot{w}$ and for function g are shown in fig.9.16(a-b). The adaptation process stops at $t = 10$ sec. It is seen that the controlled trajectory converges very fast and follows the desired trajectory very well.

For another, more complicated desired trajectory,

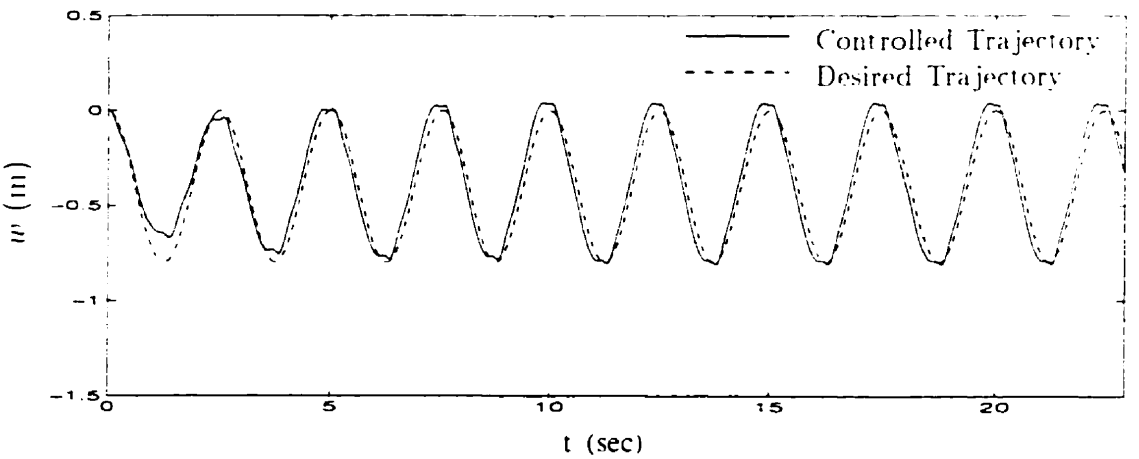
$$q(t) = 0.3\sin(0.6\pi t) - 0.3\sin(0.8\pi t) - 0.6 \ . \tag{9.294}$$

the command voltage, V_m, and the controlled trajectory, w, are shown in fig.9.17(a-b), and the estimations for variable $\dot{w}$ and for function g are shown in fig.9.18(a-b). The adaptation process also stops at $t = 10$ sec. Again, the controlled trajectory converges very fast and follows the desired trajectory very well.

These experiments demonstrate the effectiveness of the DFLS adaptive control approach in dealing with nonlinear, ill-defined systems, and reveal its potential usefulness for practical applications.



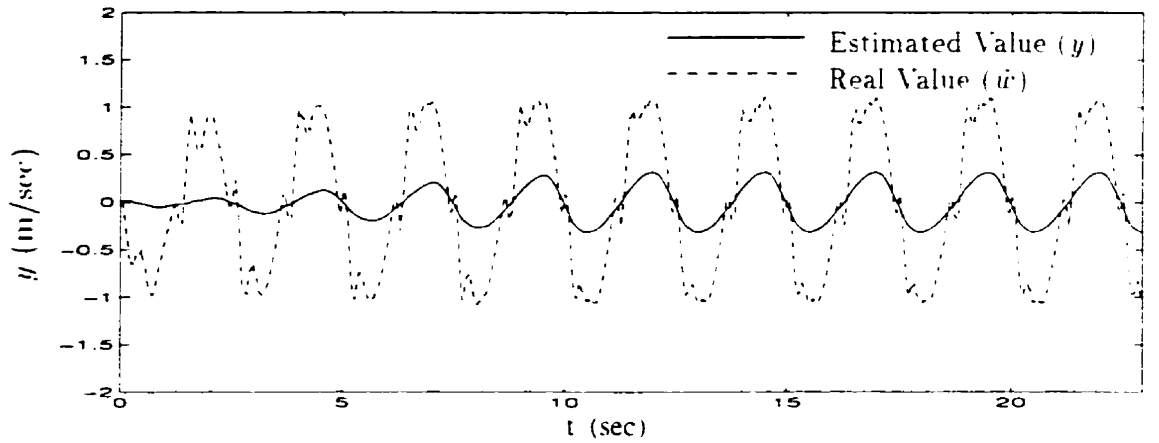
(a) Controller Output - Command Voltage



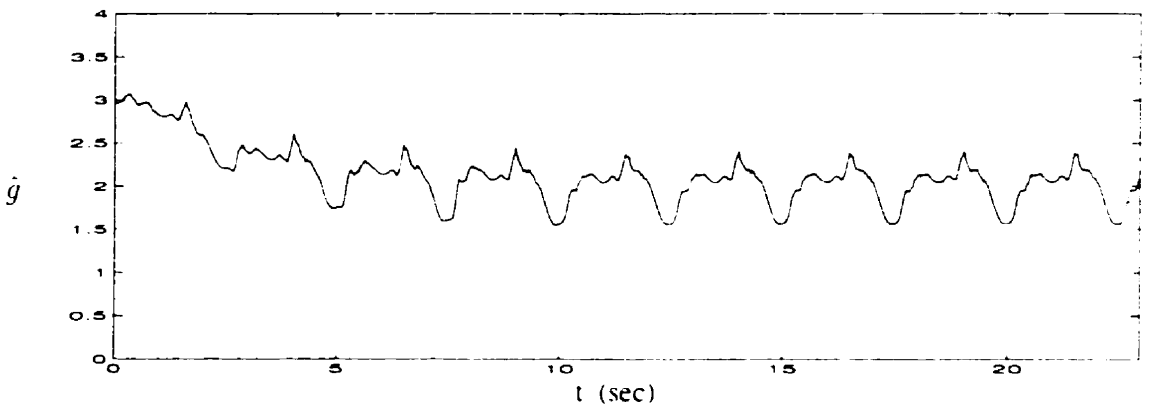
(b) Controlled Link Tip Trajectory - w

(Adaptation Terminates at t=10 sec)

Figure 9.15: Trajectory Control of a Mechanical Manipulator — Experiment 1



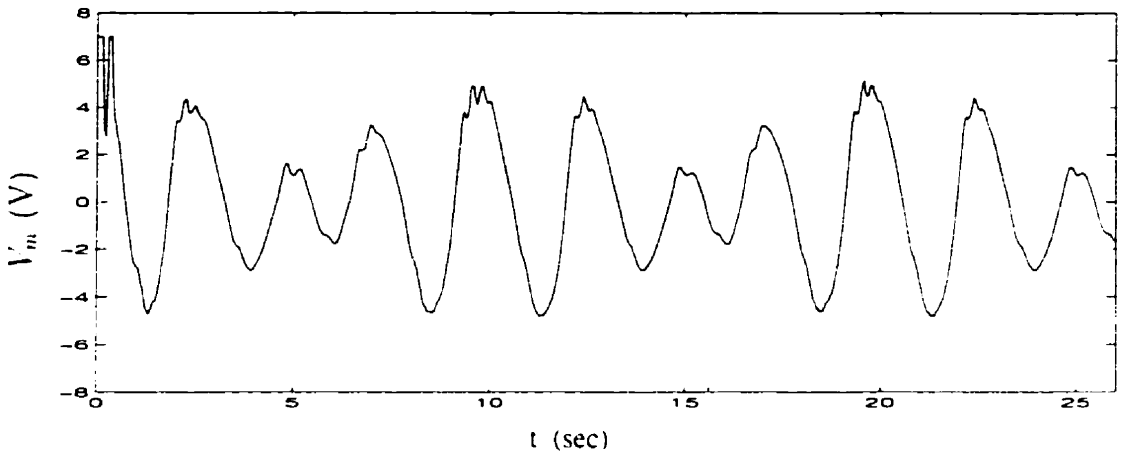
(a) Identifier Output - Link Tip Velocity



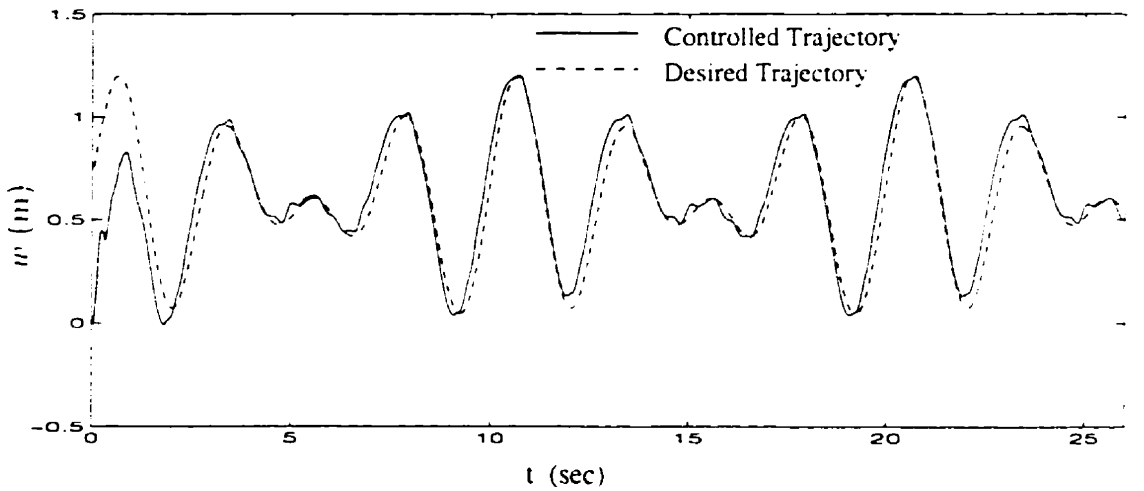
(b) Estimation for g

(Adaptation Terminates at t=10 sec)

Figure 9.16: Estimation of $\dot{w}$ and g — Experiment 1



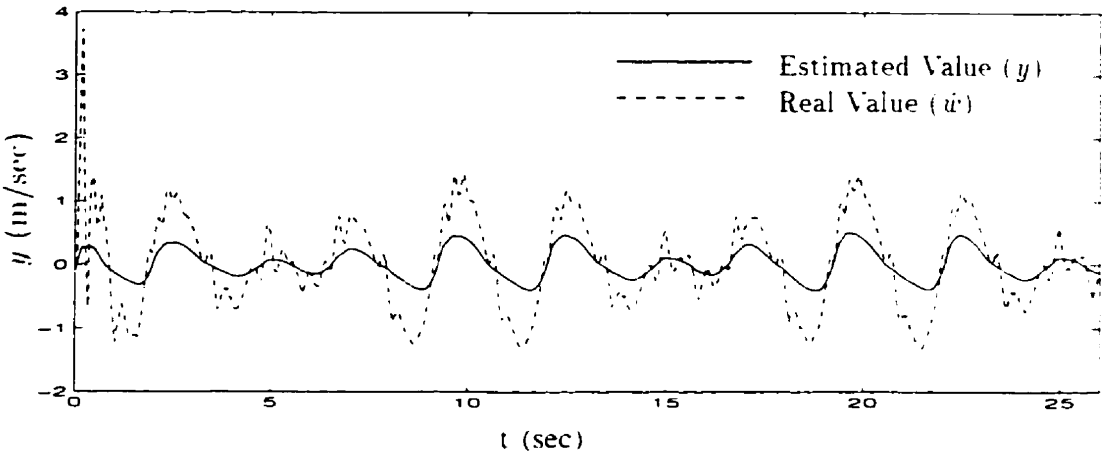
(a) Controller Output - Command Voltage



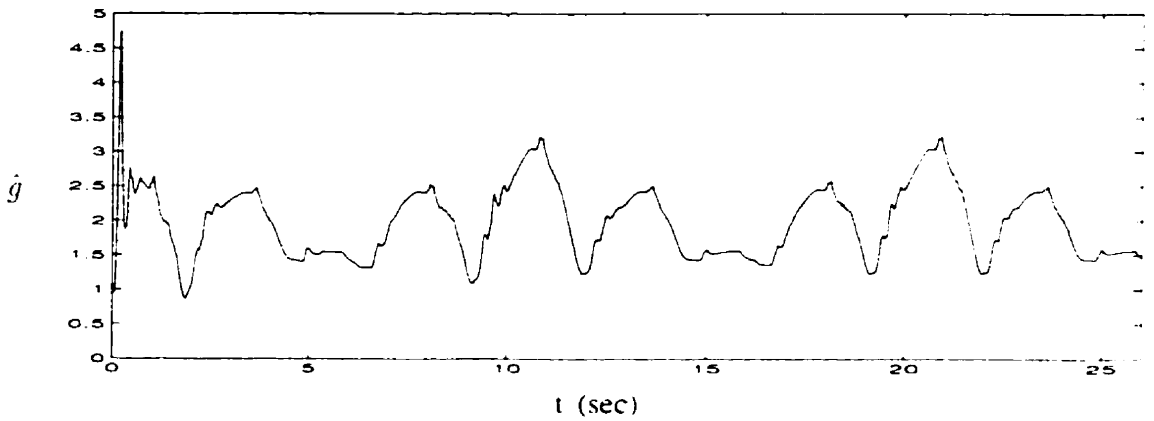
(b) Controlled Link Tip Trajectory - w

(Adaptation Terminates at $t=10$ sec)

Figure 9.17: Trajectory Control of a Mechanical Manipulator — Experiment 2



(a) Identifier Output - Link Tip Velocity



(b) Estimation for g

(Adaptation Terminates at t=10 sec)

Figure 9.18: Estimation of $\dot{w}$ and g — Experiment 2

9.11 Concluding Remarks

1. A DFLS based indirect control scheme was developed here via a Lyapunov synthesis approach for a class of nonlinear systems in companion form, eq.(9.1). Closed loop system performance and stability properties are analysed and summarized in the form of a theorem and corollaries, which theoretically justifies the fact that this control scheme enables us to achieve satisfactory performance, even under rather stringent conditions.

2. The DFLS based adaptive control algorithm was applied to control a variety of nonlinear systems, and the design procedures of DFLS controllers were illustrated in detail. In all the situations, satisfactory results were obtained, and the effectiveness of DFLS controllers in dealing with complex, nonlinear systems was demonstrated.

   Notably, the experimental results were very satisfactory from applying the DFLS adaptive controller to control the trajectory of the tip of a flexible beam manipulator, which is a very challenging mechanical system for identification and control. This further demonstrated the effectiveness of the DFLS adaptive control approach in dealing with nonlinear, ill-defined systems, and revealed its potential usefulness in practical applications.

3. In the DFLS based adaptive control algorithm, a measurable state variable, x_N, was identified, instead of identifying the unknown function, $f(\mathbf{x})$, as in the case of static FLS based control algorithms [154, 184]. It is easier to acquire knowledge of the dynamic behavior of a measurable state variable than to acquire knowledge about the unknown function, $f(\mathbf{x})$, which is not directly accessible to us.

   Also, in the DFLS based control algorithm, the system control input, u, is

explicitly included as an input of the DFLS, where any expert knowledge about the relationship between the system input, u, and the system state, x_N, can be explicitly incorporated. This is in contrast to the situation of the static FLS based control algorithm [154, 184], where u is not explicitly included as an input of the FLS.

4. Two approaches were presented to estimate the unknown function, $g(\mathbf{x})$, in eq.(9.1). One is based on a traditional self-tuning scheme, the other is a FLS approach.

   From simulation examples it is observed that the estimation of the unknown function, g, by both approaches results in comparable system performance in the situation where no expert knowledge about g is available.

   Generally, if there is no expert knowledge about g available, the former approach may be more appropriate since it is more straight forward and computationally simpler, whereas if there is such expert knowledge available, the FLS approach may be better because it can explicitly incorporate this knowledge and result in better initial values.

5. In eq.(9.1), if the unknown function, $g(\mathbf{x})$, is estimated with a FLS, Assumption 9.2 is required to ensure the system stability properties specified in Theorem 9.1. Since ε_g is a very small number and $M_{\overline{\mathbf{G}}}$ is a very large number, this assumption is not unreasonable. The author has never encountered the situation where this assumption is violated. But, the elements of the parameter vector, $\overline{\mathbf{G}}$, are not restricted to have the same sign as $g(\mathbf{x})$, as was the case in [184], which would otherwise reduce the search region of the elements of $\overline{\mathbf{G}}$ by half and could result in a poorer set of parameters.

   If Assumption 9.2 is violated in practice, then, it is necessary to either switch to the self-tuning scheme which does not require this assumption to estimate g.

or to change the initial values of $\overline{\mathbf{G}}$ and other off-line determined parameters, and repeat the training process.

6. The development of the DFLS based indirect control scheme consists of off-line design (for Θ, etc.), and on-line adaptation (for $\overline{\mathbf{Y}}$). In off-line design, the number of primary fuzzy sets for each universe of discourse, as well as the positions and shapes of membership functions for each primary fuzzy set are to be determined. Thus far, *ad hoc* analysis based on physical intuition is required. This currently is an active but immature area of research. The same is also true for α, $M_{\overline{\mathbf{Y}}}$ and other off-line determined parameters, for which explicit procedures for determining the optimal values have yet to be developed. After off-line design, the parameter vector $\overline{\mathbf{Y}}$ is adjusted on-line with the adaptive law given in eq.(9.32). If satisfactory results are not obtained by adjusting $\overline{\mathbf{Y}}$, one must return to the off-line design process to modify the off-line designed parameters and repeat the entire procedure. Furthermore, the conditions for *adequate training* of a DFLS are not discussed here, and are left as a subject for future investigation.

7. The importance of initial values in nonlinear problems is well known. In control system design, if some expert knowledge about the system is available and is described in the form of linguistic IF-THEN rules, often the situation in practice, the initial values of the parameter vector $\overline{\mathbf{Y}}$ can be assigned accordingly. These initial values are generally better than those randomly assigned, and in turn, the system may require less training and display improved behavior as demonstrated in the last chapter. In the absence of human expertise, there remains no alternative to random selection. Nevertheless, closed loop system performance will be as good as specified in Theorem 9.1.

8. An important problem in practical applications is how to deal with system

and measurement noises, which is not discussed in this work. This remains for future investigation.

9. In the illustrative application examples, the off-line design parameters were selected quite arbitrarily, without a great deal of effort, and they are by no means considered to be the optimal. This poses more stringent demands on our identification and control algorithms.

Chapter 10

The Recurrent Dynamic Fuzzy Logic System and Nonlinear System Identification

10.1 Preliminary

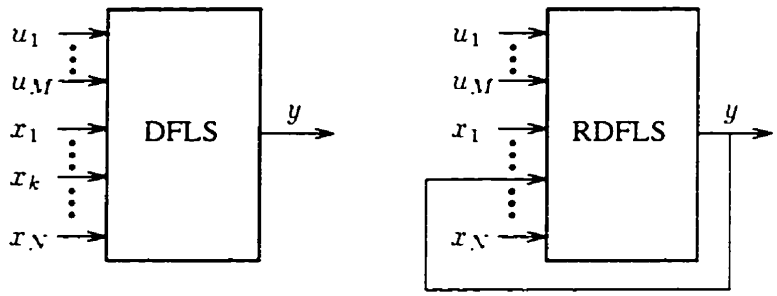
In this chapter, a type of "recurrent" DFLS is introduced and its application in nonlinear system identification demonstrated. As in the previous chapters, we restrict our scope to single output fuzzy logic systems.

By a recurrent DFLS, or RDFLS, we mean a kind of DFLS whose output is used to replace a plant state as one of its inputs, as illustrated in fig.10.1, where $\mathbf{x} \triangleq \{x_1, \ldots, x_N\}^T \in \mathcal{X} \subset R^N$ is the plant state vector, $\mathbf{u} \triangleq \{u_1, \ldots, u_M\}^T \in \mathcal{U} \subset R^M$ is the plant external input vector, and y is the estimate of the state variable, x_k, $1 \leq k \leq N$.

The mathematical expression of this system is

$$\dot{y} = -\alpha y + \boldsymbol{\Theta}^T(\mathbf{w})\overline{\mathbf{Y}} \ , \tag{10.1}$$

where $\mathbf{w} \triangleq \{w_1, \ldots, w_P\}^T \triangleq \{x_1, \ldots, x_{k-1}, y, x_{k+1}, \ldots, x_N, u_1, \ldots, u_M\}^T$, i.e., $w_i \triangleq$



Non-recurrent DFLS Recurrent DFLS

Figure 10.1: Concept of RDFLS

y, $1 \leq k \leq N$. $\boldsymbol{\Theta}$ and $\overline{\mathbf{Y}}$ are the same as those in eqs.(8.3-8.4), i.e.,

$$\begin{cases} \boldsymbol{\Theta}(\mathbf{w}) \triangleq \{\theta_1(\mathbf{w}), \ldots, \theta_I(\mathbf{w})\}^T , \\ \overline{\mathbf{Y}} \triangleq \{\overline{y}_1, \ldots, \overline{y}_I\}^T , \end{cases} \tag{10.2}$$

$$\theta_i(\mathbf{w}) \triangleq \frac{\prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{w_p - \overline{w}_p^i}{\sigma_p^i})^2]}{\sum_{i=1}^{I} \prod_{p=1}^{P} \exp[-\frac{1}{2}(\frac{w_p - \overline{w}_p^i}{\sigma_p^i})^2]} . \tag{10.3}$$

$\overline{w}_p^i \in \{\overline{w}_{j_p}, j_p = 1, \ldots, J_p\}$, $\sigma_p^i \in \{\sigma_{j_p}, j_p = 1, \ldots, J_p\}$, and J_p is the number of primary fuzzy sets in the universe of discourse $\mathcal{Z}_p$ for linguistic variable w_p.

We shall show that the RDFLS of eq.(10.1) retains the capability of approximating a large class of nonlinear functions to any desired degree of accuracy.

10.2 Universal Approximation Property

Consider the same nonlinear system as in eq.(8.6), i.e.,

$$\dot{x} = f(\mathbf{z}) . \tag{10.4}$$

where $\mathbf{z} \triangleq \{\mathbf{x}, \mathbf{u}\}^T \triangleq \{z_1, \ldots, z_P\}^T \in \mathcal{Z} \subset R^P$, $P = N + M$, i.e., $\mathbf{z}$ is composed of both system states and external inputs, x is any scalar element of the system state vector, $\mathbf{x}$, and $f : R^P \rightarrow R$ is a continuous static nonlinear map defined on the compact set $\mathcal{Z}$.

Before presenting the universal approximation theorem for RDFLS, we first prove the following lemma.

Lemma 10.1 *For given* $\mathbf{\Theta}(\mathbf{w}) = \{\theta_1(\mathbf{w}), \ldots, \theta_I(\mathbf{w})\}^T$, *and* $\mathbf{w} = \{w_1, \ldots, w_P\}^T \in \mathcal{Z} \subset R^P$, *where* $\mathcal{Z}$ *is a compact set, there exist positive constants* c_i, $i = 1, \ldots, I$, *such that*

$$\| \mathbf{\Theta}(\mathbf{w}_1) - \mathbf{\Theta}(\mathbf{w}_2) \| \leq \mathbf{c} \| \mathbf{w}_1 - \mathbf{w}_2 \| . \tag{10.5}$$

where $\mathbf{w}_1 \in \mathcal{Z}$, $\mathbf{w}_2 \in \mathcal{Z}$, *and* $\mathbf{c} \triangleq \{c_1, \ldots, c_I\}^T$.

Proof:

From eq.(10.3), it is clear that, for $i = 1, \ldots, I$, $\theta_i(\mathbf{w})$ is differentiable with respect to all the elements of $\mathbf{w}$ and the derivatives are bounded. By the mean value theorem of multivariables, then, for $i = 1, \ldots, I$,

$$\begin{aligned} \theta_i(\mathbf{w}_1) - \theta_i(\mathbf{w}_2) &= \theta_i(w_{11}, \ldots, w_{1P}) - \theta_i(w_{21}, \ldots, w_{2P}) \\ &= \frac{\partial \theta_i(\xi_1, \ldots, \xi_P)}{\partial w_1}(w_{11} - w_{21}) + \cdots + \frac{\partial \theta_i(\xi_1, \ldots, \xi_P)}{\partial w_P}(w_{1P} - w_{2P}) , \end{aligned} \tag{10.6}$$

where $\mathbf{w}_1$, $\mathbf{w}_2 \in \mathcal{Z}$, and

$$\xi_p \triangleq w_{1p} + \lambda(w_{1p} - w_{2p}) , \quad 0 < \lambda < 1, \quad p = 1, \ldots, P . \tag{10.7}$$

From eq.(10.6), we have

$$\begin{aligned} |\theta_i(\mathbf{w}_1) - \theta_i(\mathbf{w}_2)| \leq & \left| \frac{\partial \theta_i(\xi_1, \ldots, \xi_P)}{\partial w_1} \right| \cdot |w_{11} - w_{21}| + \cdots \\ & + \left| \frac{\partial \theta_i(\xi_1, \ldots, \xi_P)}{\partial w_P} \right| \cdot |w_{1P} - w_{2P}| . \end{aligned} \tag{10.8}$$

Let, for $p = 1, \ldots, P$,

$$c_{ip} \triangleq \sup_{\mathbf{w} \in \mathcal{Z}} \left\{ \left| \frac{\partial \theta_i}{\partial w_p} \right| \right\} . \tag{10.9}$$

then,

$$|\theta_i(\mathbf{w}_1) - \theta_i(\mathbf{w}_2)| \leq c_{i1}|w_{11} - w_{21}| + \cdots + c_{iP}|w_{1P} - w_{2P}| . \tag{10.10}$$

and let

$$c_i' \triangleq \max_p \{c_{ip},\ p = 1, \ldots, P\} . \tag{10.11}$$

then,

$$| \theta_i(\mathbf{w}_1) - \theta_i(\mathbf{w}_2) | \leq c_i' [\ |w_{11} - w_{21}| + \cdots + |w_{1P} - w_{2P}|\ ] . \tag{10.12}$$

Since

$$\left(\frac{|w_{11} - w_{21}| + \cdots + |w_{1P} - w_{2P}|}{P} \right)^2 \leq \frac{|w_{11} - w_{21}|^2 + \cdots + |w_{1P} - w_{2P}|^2}{P} , \tag{10.13}$$

it follows that

$$\begin{aligned} |w_{11} - w_{21}| + \cdots + |w_{1P} - w_{2P}| &\leq \sqrt{P} \left(|w_{11} - w_{21}|^2 + \cdots + |w_{1P} - w_{2P}|^2 \right)^{\frac{1}{2}} \\ &= \sqrt{P}\ \| \mathbf{w}_1 - \mathbf{w}_2 \| . \end{aligned} \tag{10.14}$$

Letting $c_i \triangleq c_i' \sqrt{P}$, and using eq.(10.14) in (10.12) yields

$$| \theta_i(\mathbf{w}_1) - \theta_i(\mathbf{w}_2) | \leq c_i \| \mathbf{w}_1 - \mathbf{w}_2 \| . \tag{10.15}$$

Since this proof is generally valid for $i = 1, \ldots, I$, we have

$$| \boldsymbol{\Theta}(\mathbf{w}_1) - \boldsymbol{\Theta}(\mathbf{w}_2) | = \begin{Bmatrix} | \theta_1(\mathbf{w}_1) - \theta_1(\mathbf{w}_2) | \\ \vdots \\ | \theta_I(\mathbf{w}_1) - \theta_I(\mathbf{w}_2) | \end{Bmatrix} \leq \begin{Bmatrix} c_1 \\ \vdots \\ c_I \end{Bmatrix} \| \mathbf{w}_1 - \mathbf{w}_2 \| \tag{10.16}$$

With $\mathbf{c} \triangleq \{c_1, \ldots, c_I\}^T$, we have

$$| \boldsymbol{\Theta}(\mathbf{w}_1) - \boldsymbol{\Theta}(\mathbf{w}_2) | \leq \mathbf{c} \| \mathbf{w}_1 - \mathbf{w}_2 \| . \tag{10.17}$$

This completes the proof.

Next, we present the universal approximation theorem of RDFLS.

Theorem 10.1 *For any $\varepsilon > 0$, there exists a RDFLS in the form of eq.(10.1), with* $\mathbf{w}$ *defined on a compact set $\mathcal{Z} \subset R^P$, such that*

$$\sup_{\mathbf{z},\mathbf{w}\in\mathcal{Z}} | y - x | < \varepsilon . \tag{10.18}$$

Proof:

Rewrite eq.(10.4) as

$$\dot{x} = -\alpha x + \alpha x + f(\mathbf{z}) . \tag{10.19}$$

With eqs.(10.1) and (10.19) we have

$$\dot{\epsilon} = -\alpha\epsilon + \overline{\mathbf{Y}}^T\boldsymbol{\Theta}(\mathbf{w}) - g(\mathbf{z}) , \tag{10.20}$$

where ϵ is the identification error, and

$$\begin{cases} \epsilon & \hat{\equiv} \quad y - x . \\ g(\mathbf{z}) & \hat{\equiv} \quad \alpha x + f(\mathbf{z}) . \end{cases} \tag{10.21}$$

By the universal approximation theorem of static fuzzy logic systems [177], $\forall\ \delta > 0$, $\exists\ \overline{\mathbf{Y}}^{*T}\boldsymbol{\Theta}(\mathbf{z})$, such that

$$\sup_{z\in\mathcal{Z}} | \overline{\mathbf{Y}}^{*T}\boldsymbol{\Theta}(\mathbf{z}) - g(\mathbf{z}) | < \delta . \tag{10.22}$$

In eq.(10.20), let $\overline{\mathbf{Y}} \hat{\equiv} \overline{\mathbf{Y}}^*$, then

$$\dot{\epsilon} = -\alpha\epsilon + \overline{\mathbf{Y}}^{*T}\boldsymbol{\Theta}(\mathbf{w}) - \overline{\mathbf{Y}}^{*T}\boldsymbol{\Theta}(\mathbf{z}) - \overline{\mathbf{Y}}^{*T}\boldsymbol{\Theta}(\mathbf{z}) - g(\mathbf{z}) . \tag{10.23}$$

The solution to eq.(10.23) is

$$\begin{aligned} \epsilon(t) \;=\; & \exp(-\alpha t)\epsilon(0) + \int_0^t \exp[-\alpha(t-\tau)][\ \overline{\mathbf{Y}}^{*T}\boldsymbol{\Theta}(\mathbf{w}) - \overline{\mathbf{Y}}^{*T}\boldsymbol{\Theta}(\mathbf{z})\ ]d\tau \\ & + \int_0^t \exp[-\alpha(t-\tau)][\ \overline{\mathbf{Y}}^{*T}\boldsymbol{\Theta}(\mathbf{z}) - g(\mathbf{z})\ ]d\tau . \end{aligned} \tag{10.24}$$

or,

$$\begin{aligned} | \epsilon(t) | \;\le\; & \exp(-\alpha t) | \epsilon(0) | + \int_0^t \exp[-\alpha(t-\tau)] | \overline{\mathbf{Y}}^{*T}[\ \boldsymbol{\Theta}(\mathbf{w}) - \boldsymbol{\Theta}(\mathbf{z})\ ] | d\tau \\ & + \int_0^t \exp[-\alpha(t-\tau)] | \overline{\mathbf{Y}}^{*T}\boldsymbol{\Theta}(\mathbf{z}) - g(\mathbf{z}) | d\tau . \end{aligned} \tag{10.25}$$

It is straightforward to verify that, for $\epsilon(0) \neq 0$, $\forall\, \delta_0 > 0$ and

$$t_0(\delta_0) \in \{t:\ t > \max\{0,\ \frac{1}{\alpha}\ln[\frac{|\ \epsilon(0)\ |}{\delta_0}]\}\}\ . \tag{10.26}$$

that

$$\exp(-\alpha t_0)\cdot |\ \epsilon(0)\ | <\ \delta_0\ . \tag{10.27}$$

Referring to Lemma 10.1, then,

$$\begin{aligned} |\ \overline{\mathbf{Y}}^{*T}[\ \boldsymbol{\Theta}(\mathbf{w}) - \boldsymbol{\Theta}(\mathbf{z})\ ]\ | &\leq |\ \overline{\mathbf{Y}}^{*T}\ |\cdot |\ \boldsymbol{\Theta}(\mathbf{w}) - \boldsymbol{\Theta}(\mathbf{z})\ | \\ &\leq |\ \overline{\mathbf{Y}}^{*T}\ |\cdot |\ \mathbf{c}\ |\cdot |\ \mathbf{w} - \mathbf{z}\ |\ . \end{aligned} \tag{10.28}$$

But $\|\ \mathbf{w} - \mathbf{z}\ \| = |\ \epsilon\ |$. Let $B \triangleq |\ \overline{\mathbf{Y}}^{*T}\ |\cdot |\ \mathbf{c}\ |$, and considering eqs.(10.22) and (10.27), then,

$$\begin{aligned} |\ \epsilon(t)\ | &< \delta_0 + \int_0^t |\ \epsilon\ |\cdot B\cdot\exp[-\alpha(t-\tau)]d\tau + \int_0^t \delta\cdot\exp[-\alpha(t-\tau)]d\tau \\ &\leq \delta_0 - \frac{\delta}{\alpha} + \int_0^t |\ \epsilon\ |\cdot B\cdot\exp[-\alpha(t-\tau)]d\tau\ . \end{aligned} \tag{10.29}$$

By the Gronwall-Bellman lemma [114, Chap.2],

$$|\ \epsilon(t)\ | < (\ \delta_0 + \frac{\delta}{\alpha}\ )\cdot\exp\{\ \int_0^t B\cdot\exp[-\alpha(t-\tau)]d\tau\}\ . \tag{10.30}$$

Since

$$\int_0^t B\cdot\exp[-\alpha(t-\tau)]d\tau = \frac{B}{\alpha}\ [\ 1 - \exp(-\alpha t)\ ] \leq \frac{B}{\alpha}\ , \tag{10.31}$$

use of eq.(10.31) in (10.30) yields

$$|\ \epsilon(t)\ | < (\ \delta_0 + \frac{\delta}{\alpha}\ )\cdot\exp(\ \frac{B}{\alpha}\ )\ . \tag{10.32}$$

Let

$$\varepsilon \triangleq (\delta_0 + \frac{\delta}{\alpha})\cdot\exp(\ \frac{B}{\alpha}\ )\ . \tag{10.33}$$

then

$$| \epsilon(t) | < \varepsilon . \tag{10.34}$$

This completes the proof of Theorem 10.1.

Although this theorem reveals the universal approximation property of a RDFLS, it only indicates that a RDFLS can be a universal approximator, without providing any guidance on how one might construct such a RDFLS to approximate a given physical system. In the following, we address this latter point by developing a stable identification algorithm that is based on RDFLS, and is applicable to a large class of nonlinear systems.

10.3 RDFLS Identification of Nonlinear Systems

10.3.1 Preliminary

As in section 8.3, consider a general dynamic system of the form

$$\dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{u}) . \tag{10.35}$$

where $\mathbf{x} \stackrel{\Delta}{=} \{x_1, \ldots, x_N\}^T \in R^N$ and $\mathbf{u} \stackrel{\Delta}{=} \{u_1, \ldots, u_M\}^T \in R^M$ are the state vector and external input vector of the physical process, respectively. N and M represent the total number of states and external inputs. $F : R^{N+M} \rightarrow R^N$ is a continuous nonlinear vector function defined on a compact set $\mathcal{Z} \subset R^{N+M}$. We take assumption 8.1 to hold for this system as well.

We can rewrite eq.(10.35) into a set of state equations, i.e., for $k = 1, \ldots, N$,

$$\dot{x}_k = f_k(\mathbf{x}, \mathbf{u}) , \tag{10.36}$$

where $f_k : R^{N+M} \rightarrow R$ is a continuous nonlinear function defined on the compact set $\mathcal{Z}$. To identify the kth state, x_k, $k \in \{1, \ldots, N\}$, a RDFLS in form of eq.(10.1)

is used.

$$\dot{y}_k = -\alpha_k y_k + \mathbf{\Theta}_k^T(\mathbf{w})\overline{\mathbf{Y}}_k \ . \qquad \alpha > 0 \ . \tag{10.37}$$

where α_k is a positive constant. $\mathbf{\Theta}_k$ and $\overline{\mathbf{Y}}_k$ are as defined in eqs.(10.2-10.3), and $\mathbf{w} \triangleq \{w_1, \ldots, w_P\}^T \triangleq \{x_1, \ldots, x_{k-1}, y_k, x_{k+1}, \ldots, x_N, u_1, \ldots, u_M\}^T$, i.e., $w_k \triangleq y_k$, which is the estimate of x_k.

Again, as in the case of the DFLS in section 8.3, the RDFLS of eq.(10.37) is characterized by the free design parameters $\overline{w}_{j_p}$, σ_{j_p}, $\overline{y}_{ki}$, and α_k, where $i = 1, \ldots, I$, $p = 1, \ldots, P$, and $j_p = 1, \ldots, J_p$. Again, it is assumed that, for $p = 1, \ldots, P$, the fuzzy partition parameter J_p, the membership function parameters $\overline{w}_{j_p}$ and σ_{j_p}, as well as the parameter α are designed off-line, leaving only $\overline{\mathbf{Y}}_k = \{\overline{y}_{k1}, \ldots, \overline{y}_{kI}\}^T$ as an on-line adjustable parameter vector.

Our objective here is to develop a stable adaptive law for the RDFLS identifier such that it can identify an unknown nonlinear dynamic system in the form of eq.(10.36), with the identification error being bounded and as small as possible, ideally converging to zero, and at the same time, all the parameters should also be bounded.

10.3.2 Adaptive Law and Stability Properties

Let the identification error of the state variable, x_k, be

$$\epsilon_k \triangleq y_k - x_k \ . \tag{10.38}$$

following the well known Lyapunov synthesis approach we obtain a adaptive law for $\overline{\mathbf{Y}}_k$ as follows.

$$\dot{\overline{\mathbf{Y}}}_k = -\mathbf{H}_k\mathbf{\Theta}_k(\mathbf{w})\epsilon_k - S_k\beta_k\mathbf{H}_k\overline{\mathbf{Y}}_k \ , \quad k \in \{1, \ldots, N\} \ . \tag{10.39}$$

where $\mathbf{H}_k$ is a constant positive definite symmetric matrix. S_k is a switch defined as

$$S_k = \begin{cases} 0\ . \ \ if\ \ \| \overline{\mathbf{Y}}_k \| < M_{\overline{\mathbf{Y}}_k}\ . \\ \qquad or\ \ \| \overline{\mathbf{Y}}_k \| = M_{\overline{\mathbf{Y}}_k}\ \ and\ \ \boldsymbol{\Theta}_k^T(\mathbf{w})\mathbf{H}_k\overline{\mathbf{Y}}_k\epsilon_k \geq 0\ . \\ \qquad or\ \ \| \overline{\mathbf{Y}}_k \| > M_{\overline{\mathbf{Y}}_k}\ \ and\ \ \boldsymbol{\Theta}_k^T(\mathbf{w})\mathbf{H}_k\overline{\mathbf{Y}}_k\epsilon_k > 0\ . \\ \\ 1\ . \ \ otherwise. \end{cases} \tag{10.40}$$

$M_{\overline{\mathbf{Y}}_k}$ is a positive design parameter, and β_k is a positive design parameter that satisfies

$$\beta_k \begin{cases} \geq -\frac{\boldsymbol{\Theta}_k^T(\mathbf{w})\mathbf{H}_k\overline{\mathbf{Y}}_k}{\overline{\mathbf{Y}}_k^T\mathbf{H}_k\overline{\mathbf{Y}}_k}\epsilon_k\ . \ \ if\ \ \| \overline{\mathbf{Y}}_k \| = M_{\overline{\mathbf{Y}}_k}\ \ and\ \ \boldsymbol{\Theta}_k^T(\mathbf{w})\mathbf{H}_k\overline{\mathbf{Y}}_k\epsilon_k < 0\ . \\ \\ > -\frac{\boldsymbol{\Theta}_k^T(\mathbf{w})\mathbf{H}_k\overline{\mathbf{Y}}_k}{\overline{\mathbf{Y}}_k^T\mathbf{H}_k\overline{\mathbf{Y}}_k}\epsilon_k\ . \ \ if\ \ \| \overline{\mathbf{Y}}_k \| > M_{\overline{\mathbf{Y}}_k}\ \ and\ \ \boldsymbol{\Theta}_k^T(\mathbf{w})\mathbf{H}_k\overline{\mathbf{Y}}_k\epsilon_k \leq 0\ . \end{cases} \tag{10.41}$$

The stability properties of this identifier are summarized in the following theorem.

Theorem 10.2 *Consider an unknown nonlinear dynamic system in form of eq.(10.36), which is to be identified by the RDFLS of eq.(10.37) by adjusting the parameter vector $\overline{\mathbf{Y}}_k$ of the RDFLS with the adaptive law of eq.(10.39). The RDFLS identifier satisfies the following properties.*

10.2.1 $\| \overline{\mathbf{Y}}_k \| \leq M_{\overline{\mathbf{Y}}_k}$;

10.2.2 $| \epsilon_k | \leq M_{\epsilon_k}$, *where* M_{ϵ_k} *is a positive constant;*

10.2.3 $\int_0^t \epsilon_k^2(\tau)d\tau \leq a + b\int_0^t r_k^2(\tau)d\tau$, *where* a *and* b *are constants, and if* $r_k(t) \in \mathcal{L}_2[0,\infty)$, *i.e.*, $(\int_0^\infty \| r_k(\tau) \|^2 d\tau)^{\frac{1}{2}} < \infty$, *then,* $\lim_{t\to\infty} | \epsilon_k(t) | = 0$.

10.3.3 Proof of Theorem 10.2

A Proof of Theorem 10.2.1

This is similar to the proof of Theorem 8.2.1. and therefore. it is omitted here.

B Proof of Theorem 10.2.2

Let

$$r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k) \triangleq \boldsymbol{\Theta}_k^T(\mathbf{x}, \mathbf{u})\overline{\mathbf{Y}}_k - \alpha_k x_k - f_k(\mathbf{x}, \mathbf{u}) . \tag{10.42}$$

Rewrite eq.(10.36) as

$$\dot{x}_k = -\alpha_k x_k + \alpha_k x_k + f_k(\mathbf{x}, \mathbf{u}) . \tag{10.43}$$

and rewrite eq.(10.37) as

$$\begin{aligned} \dot{y}_k &= -\alpha_k y_k - \boldsymbol{\Theta}_k^T(\mathbf{w})\overline{\mathbf{Y}}_k - \boldsymbol{\Theta}_k^T(\mathbf{w})\overline{\mathbf{Y}}_k^* + \boldsymbol{\Theta}_k^T(\mathbf{w})\overline{\mathbf{Y}}_k^* \\ &\quad - \boldsymbol{\Theta}_k^T(\mathbf{x}, \mathbf{u})\overline{\mathbf{Y}}_k^* + \boldsymbol{\Theta}_k^T(\mathbf{x}, \mathbf{u})\overline{\mathbf{Y}}_k^* . \end{aligned} \tag{10.44}$$

where $\overline{\mathbf{Y}}_k^*$ is an optimal parameter vector defined in eq.(8.30) which minimizes the static modeling error, $r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k^*)$. Referring to Lemma 8.1 we have

$$\begin{cases} \| \overline{\mathbf{Y}}_k^* \| \leq M_{\overline{\mathbf{Y}}_k} . \\ \sup_{\{\mathbf{x},\mathbf{u}\}\in\mathcal{Z}} | r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k^*) | \leq M_{r_k} . \end{cases} \tag{10.45}$$

where M_{r_k} is a positive constant. Considering eqs.(10.43-10.44) gives

$$\dot{\epsilon}_k = -\alpha_k \epsilon_k + \boldsymbol{\Theta}_k^T(\mathbf{w})\boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} + \overline{\mathbf{Y}}_k^{*T}\boldsymbol{\Delta}_{\boldsymbol{\Theta}_k} + r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k^*) . \tag{10.46}$$

where

$$\begin{cases} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \triangleq \overline{\mathbf{Y}}_k - \overline{\mathbf{Y}}_k^* . \\ \boldsymbol{\Delta}_{\boldsymbol{\Theta}_k} \triangleq \boldsymbol{\Theta}_k(\mathbf{w}) - \boldsymbol{\Theta}_k(\mathbf{z}) . \end{cases} \tag{10.47}$$

and $\mathbf{z} = \{\mathbf{x}, \mathbf{u}\}^T$.

Consider the Lyapunov function candidate

$$V = \frac{1}{2}\epsilon_k^2 + \frac{1}{2}\boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k}^T \mathbf{H}_k^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \ . \tag{10.48}$$

Its derivative is

$$\dot{V} = \epsilon_k \dot{\epsilon}_k + \dot{\boldsymbol{\Delta}}_{\overline{\mathbf{Y}}_k}^T \mathbf{H}_k^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \ . \tag{10.49}$$

Using eq.(10.46) in (10.49) and considering $\dot{\boldsymbol{\Delta}}_{\overline{\mathbf{Y}}_k} = \dot{\overline{\mathbf{Y}}}_k$ results in

$$\begin{aligned} \dot{V} \ &= \ -\alpha_k \epsilon_k^2 + \boldsymbol{\Theta}_k^T(\mathbf{w}) \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \epsilon_k + \overline{\mathbf{Y}}_k^{*T} \boldsymbol{\Delta}_{\boldsymbol{\Theta}_k} \epsilon_k + r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k^*) \epsilon_k \\ &\quad + \dot{\overline{\mathbf{Y}}}_k^T \mathbf{H}_k^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \ . \end{aligned} \tag{10.50}$$

The adaptive law, eq.(10.39), in eq.(10.50) yields

$$\dot{V} = -\alpha_k \epsilon_k^2 + r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k^*) \epsilon_k + \overline{\mathbf{Y}}_k^{*T} \boldsymbol{\Delta}_{\boldsymbol{\Theta}_k} \epsilon_k - S_k \beta_k \overline{\mathbf{Y}}_k^T \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \ . \tag{10.51}$$

Since

$$\overline{\mathbf{Y}}_k^T \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} = \frac{1}{2} \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 - \frac{1}{2} \| \overline{\mathbf{Y}}_k \|^2 - \frac{1}{2} \| \overline{\mathbf{Y}}_k^* \|^2 \ . \tag{10.52}$$

and for $S_k = 1$, $\| \overline{\mathbf{Y}}_k \| \geq M_{\overline{\mathbf{Y}}_k}$, therefore,

$$S_k \beta_k \overline{\mathbf{Y}}_k^T \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \geq \frac{1}{2} S_k \beta_k \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 \geq 0 \ . \tag{10.53}$$

Use of eq.(10.53) in (10.51) yields

$$\begin{aligned} \dot{V} \ &\leq \ -\alpha_k \epsilon_k^2 - \frac{1}{2} S_k \beta_k \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 + r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k^*) \epsilon_k + \overline{\mathbf{Y}}_k^{*T} \boldsymbol{\Delta}_{\boldsymbol{\Theta}_k} \epsilon_k \\ &= \ -\frac{1}{2}\alpha_k \epsilon_k^2 - \frac{1}{2} \beta_k \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 + \frac{1}{2}(1 - S_k) \beta_k \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 \\ &\quad - \frac{1}{4}\alpha_k \epsilon_k^2 + r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k^*) \epsilon_k - \frac{1}{4}\alpha_k \epsilon_k^2 + \overline{\mathbf{Y}}_k^{*T} \boldsymbol{\Delta}_{\boldsymbol{\Theta}_k} \epsilon_k \ . \end{aligned} \tag{10.54}$$

Reorganizing individual terms in eq.(10.54) produces

$$\begin{aligned} \frac{1}{2}(1 - S_k) \beta_k \| \boldsymbol{\Delta}_k \|^2 \ &\leq \ \frac{1}{2} \beta_k \| \boldsymbol{\Delta}_k \|^2 \leq \ \frac{1}{2} \beta_k (\| \overline{\mathbf{Y}}_k \| + \| \overline{\mathbf{Y}}_k^* \|)^2 \\ &\leq \ 2 \beta_k M_{\overline{\mathbf{Y}}_k}^2 \ . \end{aligned} \tag{10.55}$$

$$
\begin{aligned}
-\frac{1}{4}\alpha_k\epsilon_k^2 + r_k(\mathbf{x},\mathbf{u},\boldsymbol{\Theta}_k,\overline{\mathbf{Y}}_k^*)\epsilon_k &= -\frac{\alpha_k}{4}\Big[\,\epsilon_k - \frac{2}{\alpha_k}r_k(\mathbf{x},\mathbf{u},\boldsymbol{\Theta}_k,\overline{\mathbf{Y}}_k^*)\,\Big]^2 \\
&\quad + \frac{1}{\alpha_k}r_k^2(\mathbf{x},\mathbf{u},\boldsymbol{\Theta}_k,\overline{\mathbf{Y}}_k^*) \\
&\le \frac{1}{\alpha_k}r_k^2(\mathbf{x},\mathbf{u},\boldsymbol{\Theta}_k,\overline{\mathbf{Y}}_k^*) \\
&\le \frac{1}{\alpha_k}M_{r_k}^2 . \qquad (10.56)
\end{aligned}
$$

$$
\begin{aligned}
-\frac{1}{4}\alpha_k\epsilon_k^2 + \overline{\mathbf{Y}}_k^{*T}\boldsymbol{\Delta}_{\boldsymbol{\Theta}_k}\epsilon_k &= -\frac{\alpha_k}{4}\Big[\,\epsilon_k - \frac{2}{\alpha_k}(\overline{\mathbf{Y}}_k^{*T}\boldsymbol{\Delta}_{\boldsymbol{\Theta}_k})\,\Big]^2 + \frac{1}{\alpha_k}(\overline{\mathbf{Y}}_k^{*T}\boldsymbol{\Delta}_{\boldsymbol{\Theta}_k})^2 \\
&\le \frac{1}{\alpha_k}(\overline{\mathbf{Y}}_k^{*T}\boldsymbol{\Delta}_{\boldsymbol{\Theta}_k})^2 \\
&\le \frac{1}{\alpha_k}(|\,\overline{\mathbf{Y}}_k^{*T}\,|\cdot|\,\boldsymbol{\Delta}_{\boldsymbol{\Theta}_k}\,|)^2 . \qquad (10.57)
\end{aligned}
$$

Since $0 \le \theta_{ki}(\mathbf{w}) \le 1$, and $0 \le \theta_{ki}(\mathbf{z}) \le 1$, $i = 1,\ldots,I$, we have,

$$
|\,\theta_{ki}(\mathbf{w}) - \theta_{ki}(\mathbf{z})\,| \le 1 . \qquad (10.58)
$$

therefore,

$$
\begin{aligned}
|\,\overline{\mathbf{Y}}_k^{*T}\,|\cdot|\,\boldsymbol{\Delta}_{\boldsymbol{\Theta}_k}\,| &= |\,\overline{\mathbf{Y}}_k^{*T}\,|\cdot|\,\boldsymbol{\Theta}_k(\mathbf{w}) - \boldsymbol{\Theta}_k(\mathbf{z})\,| \\
&= \sum_{i=1}^{I}|\,\overline{y}_{ki}^*\,|\cdot|\,\theta_{ki}(\mathbf{w}) - \theta_{ki}(\mathbf{z})\,| \\
&\le \sum_{i=1}^{I}|\,\overline{y}_{ki}^*\,| . \qquad (10.59)
\end{aligned}
$$

Since

$$
\Big(\frac{\sum_{i=1}^{I}|\,\overline{y}_{ki}^*\,|}{I}\Big)^2 \le \frac{\sum_{i=1}^{I}|\,\overline{y}_{ki}^*\,|^2}{I} . \qquad (10.60)
$$

or,

$$
\sum_{i=1}^{I}|\,\overline{y}_{ki}^*\,| \le \sqrt{I}\,\|\,\overline{\mathbf{Y}}_k^*\,\| \le \sqrt{I}M_{\overline{\mathbf{Y}}_k} . \qquad (10.61)
$$

so that,

$$
-\frac{1}{4}\alpha_k\epsilon_k^2 + \overline{\mathbf{Y}}_k^{*T}\boldsymbol{\Delta}_{\boldsymbol{\Theta}_k}\epsilon_k \le \frac{I}{\alpha_k}M_{\overline{\mathbf{Y}}_k}^2 . \qquad (10.62)
$$

Also,

$$-\frac{1}{2}\beta_k \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 \leq -\frac{1}{2}\beta_k \frac{\boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k}^T \mathbf{H}_k^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k}}{\lambda_{\max}(\mathbf{H}_k^{-1})} . \tag{10.63}$$

where $\lambda_{\max}(\mathbf{H}_k^{-1})$ is the maximum eigenvalue of matrix $\mathbf{H}_k^{-1}$.

Using eqs.(10.55-10.56) and (10.62-10.63) in eq.(10.54) yields

$$\begin{aligned} \dot{V} \leq & -\frac{1}{2}\alpha_k \epsilon_k^2 - \frac{1}{2}\frac{\beta_k}{\lambda_{\max}(\mathbf{H}_k^{-1})} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k}^T \mathbf{H}_k^{-1} \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} + 2\beta_k M_{\overline{\mathbf{Y}}_k}^2 + \frac{I}{\alpha_k} M_{\overline{\mathbf{Y}}_k}^2 \\ & + \frac{1}{\alpha_k} M_{r_k}^2 \end{aligned} \tag{10.64}$$

Let

$$\kappa \triangleq \min\{ \alpha_k, \frac{\beta_k}{\lambda_{\max}(\mathbf{H}_k^{-1})} \} . \tag{10.65}$$

$$C \triangleq 2\beta_k M_{\overline{\mathbf{Y}}_k}^2 + \frac{I}{\alpha_k} M_{\overline{\mathbf{Y}}_k}^2 - \frac{1}{\alpha_k} M_{r_k}^2 . \tag{10.66}$$

then

$$\dot{V} \leq -\kappa V + C . \tag{10.67}$$

Therefore,

$$\dot{V} \leq 0 , \quad if \quad V \geq \frac{C}{\kappa} . \tag{10.68}$$

Thus, if $V < \frac{C}{\kappa}$, then V is bounded, implying boundedness of ϵ_k and $\boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k}$. If $V \geq \frac{C}{\kappa}$, then eq.(10.68) indicates the boundedness of V, again implying the boundedness of ϵ_k and $\boldsymbol{\Delta}_k$. Therefore, there exists a positive constant, M_{ϵ_k}, such that $| \epsilon_k | \leq M_{\epsilon_k}$. This completes the proof of Theorem 10.2.2.

C Proof of Theorem 10.2.3

Considering eq.(10.54) we have

$$\begin{aligned} \dot{V} &\leq -\alpha_k \epsilon_k^2 - \frac{1}{2} S_k \beta_k \| \boldsymbol{\Delta}_{\overline{\mathbf{Y}}_k} \|^2 + r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k^*)\epsilon_k + \overline{\mathbf{Y}}_k^{*T} \boldsymbol{\Delta}_{\boldsymbol{\Theta}_k} \epsilon_k \\ &\leq -\alpha_k \epsilon_k^2 + r_k(\mathbf{x}, \mathbf{u}, \boldsymbol{\Theta}_k, \overline{\mathbf{Y}}_k^*)\epsilon_k + \overline{\mathbf{Y}}_k^{*T} \boldsymbol{\Delta}_{\boldsymbol{\Theta}_k} \epsilon_k . \end{aligned} \tag{10.69}$$

Substitution of eqs.(10.56–10.57) in eq.(10.69) gives

$$\dot{V} \leq -\frac{1}{2}\alpha_k \epsilon_k^2 + \frac{1}{\alpha_k} r_k^2(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}_k^*) + \frac{1}{\alpha_k}(\mid \overline{\mathbf{Y}}_k^{*T} \mid \cdot \mid \mathbf{\Delta}_{\mathbf{\Theta}_k} \mid)^2 . \tag{10.70}$$

By Lemma 10.1 we have

$$\mid \mathbf{\Delta}_{\mathbf{\Theta}_k} \mid = \mid \mathbf{\Theta}_k(\mathbf{w}) - \mathbf{\Theta}_k(\mathbf{z}) \mid \leq \mathbf{c} \parallel \mathbf{w} - \mathbf{z} \parallel . \tag{10.71}$$

or,

$$\mid \mathbf{\Delta}_{\mathbf{\Theta}_k} \mid \leq \mathbf{c} \mid \epsilon_k \mid . \tag{10.72}$$

where $\mathbf{c} \triangleq \{c_1, \ldots, c_l\}^T$, whose elements are positive constants. Use of eq.(10.72) in (10.70) and letting $B_k \triangleq (\mid \overline{\mathbf{Y}}_k^{*T} \mid \cdot \mathbf{c})$ results in

$$\dot{V} \leq -(\frac{1}{2}\alpha_k - \frac{B_k^2}{\alpha_k})\epsilon_k^2 + \frac{1}{\alpha_k} r_k^2(\mathbf{x}, \mathbf{u}, \mathbf{\Theta}_k, \overline{\mathbf{Y}}_k^*). \tag{10.73}$$

Integrating both sides of eq.(10.73) yields

$$V(t) - V(0) \leq -\frac{\alpha_k^2 - 2B_k^2}{2\alpha_k} \int_0^t \epsilon_k^2(\tau) d\tau + \frac{1}{\alpha_k} \int_0^t r_k^2(\tau) d\tau . \tag{10.74}$$

Since α_k is a design parameter, we can choose $\alpha_k^2 > 2B_k^2$. Let

$$a \triangleq \frac{2\alpha_k}{\alpha_k^2 - 2B_k^2} \sup_{t>0}\{V(0) - V(t)\} , \tag{10.75}$$

$$b \triangleq \frac{2}{\alpha_k^2 - 2B_k^2} . \tag{10.76}$$

then

$$\int_0^t \epsilon_k^2(\tau) d\tau \leq a + b \int_0^t r_k^2(\tau) d\tau , \tag{10.77}$$

where a and b are constants.

If $r_k(t) \in \mathcal{L}_2$, then by eq.(10.77), $\epsilon_k(t) \in \mathcal{L}_2$. By Theorem 10.2.2, we know that $\epsilon_k(t)$ is bounded, i.e., $\epsilon_k(t) \in \mathcal{L}_\infty$. Therefore, $\epsilon_k(t) \in (\mathcal{L}_2 \bigcap \mathcal{L}_\infty)$. From eq.(10.46), the boundedness of $\dot{\epsilon}_k$ can be concluded, since all the terms on the right hand side

of the equation are bounded. That is, $\dot{\epsilon}_k(t) \in \mathcal{L}_\infty$. By Corollary 2.9 in [114], we conclude that

$$\lim_{t\to\infty} | \epsilon_k(t) | = 0 . \tag{10.78}$$

This completes the proof of Theorem 10.2.

10.4 Illustrative Applications – Identification of Nonlinear Systems

10.4.1 Example 1 – Identification of a Simple Nonlinear System

Consider the following nonlinear system.

$$\dot{x} = \frac{\cos(x)}{1+x^2} + u . \tag{10.79}$$

Our objective here is to identify the state variable, x, using the RDFLS identification algorithm developed in previous sections. The dynamic model will be treated strictly as a black box, and it is unknown to the identifier.

Let

$$\mathbf{w} \triangleq \{w_1, w_2\}^T \triangleq \{y, u\}^T . \tag{10.80}$$

A RDFLS identifier, F, is used to identify x, i.e.,

$$y = F(\mathbf{w}') , \tag{10.81}$$

where $\mathbf{w}'$ is the filtered value of $\mathbf{w}$, with the prefilters defined as

$$\begin{cases} w_1' &= f_1(w_1) \triangleq (w_1 - 4)\cdot\frac{3}{4} , \\ w_2' &= f_2(w_2) \triangleq w_2 \cdot 3 . \end{cases} \tag{10.82}$$

y is the estimate of x, and is specified by eq.(10.1).

$$\dot{y} = -\alpha y + \mathbf{\Theta}^T(\mathbf{w}')\overline{\mathbf{Y}} \ . \tag{10.83}$$

The estimation error, ϵ, is defined as

$$\epsilon \triangleq y - x \ . \tag{10.84}$$

F has two inputs, w_1' and w_2', and one output, y. In both universes of discourse of input variables, seven primary fuzzy sets, A_{1j_1} and A_{2j_2}, $j_1 = 1,\ldots,7$, $j_2 = 1,\ldots,7$, are defined. Gaussian type membership functions are used for the primary fuzzy sets.

$$\begin{aligned}
\mu_{A_{p1}}(w_p') &\triangleq \begin{cases} 1 \ , & if \quad w_p' < \overline{w}_{p1} \ . \\ \exp[-\frac{1}{2}(\frac{w_p' - \overline{w}_{p1}}{\sigma_{p1}})^2] \ , & if \quad w_p' \geq \overline{w}_{p1} \ . \end{cases} \\
\mu_{A_{pj_p}}(w_p') &\triangleq \exp[-\frac{1}{2}(\frac{w_p' - \overline{w}_{pj_p}}{\sigma_{pj_p}})^2] \ , \quad j_1, j_2 = 2,\ldots,6. \\
\mu_{A_{p7}}(w_p') &\triangleq \begin{cases} \exp[-\frac{1}{2}(\frac{w_p' - \overline{w}_{p7}}{\sigma_{p7}})^2] \ , & if \quad w_p' < \overline{w}_{p7} \ . \\ 1 \ , & if \quad w_p' \geq \overline{w}_{p7} \ . \end{cases}
\end{aligned} \tag{10.85}$$

Let shape parameters of all the primary fuzzy sets be 0.45, i.e.,

$$\sigma_{pj_p} \triangleq 0.45 \ . \tag{10.86}$$

Define position parameters, $\overline{w}_{pj_p}$, as, for $p = 1, 2$,

$$\{\overline{w}_{p1}, \ \overline{w}_{p2}, \ \overline{w}_{p3}, \ \overline{w}_{p4}, \ \overline{w}_{p5}, \ \overline{w}_{p6}, \ \overline{w}_{p7}\}^T \triangleq \{-3, \ -2, \ -1, \ 0, \ 1, \ 2, \ 3\}^T \ . \tag{10.87}$$

In this RDFLS, eq.(10.83),

$$\begin{cases} \mathbf{\Theta}(\mathbf{w}') \triangleq \{\theta_1(\mathbf{w}'), \ \theta_2(\mathbf{w}'), \ \ldots \ \theta_{49}(\mathbf{w}')\}^T \ . \\ \overline{\mathbf{Y}} \triangleq \{\overline{y}_1, \ \overline{y}_2, \ \ldots \ \overline{y}_{49}\}^T \ . \end{cases} \tag{10.88}$$

and

$$\theta_i^1(\mathbf{w}') = \frac{\prod_{p=1}^{2} \mu_{A_p^i}(w_p')}{\sum_{i=1}^{49} \prod_{p=1}^{2} \mu_{A_p^i}(w_p')} \ . \tag{10.89}$$

and for $i = 1, \ldots, 49$, $p = 1, 2$.

$$A_p^i \in \{A_{pj_p} : j_p = 1, \ldots, 7\} . \tag{10.90}$$

The free parameter vector, $\overline{\mathbf{Y}}$, is adjusted with the adaptive law, eq.(10.39), i.e.,

$$\dot{\overline{\mathbf{Y}}} = -\mathbf{H}\boldsymbol{\Theta}(\mathbf{w}')\epsilon - S\beta\mathbf{H}\overline{\mathbf{Y}} . \tag{10.91}$$

In computer implementation, the derivative is approximated by a difference, therefore,

$$\overline{\mathbf{Y}}(kT + T) = \overline{\mathbf{Y}}(kT) - T\left[\mathbf{H}\boldsymbol{\Theta}(\mathbf{w}'(kT))\epsilon(kT) + S\beta\mathbf{H}\overline{\mathbf{Y}}(kT) \right] , \tag{10.92}$$

where T is the time incremental step and is set at 0.04 sec.

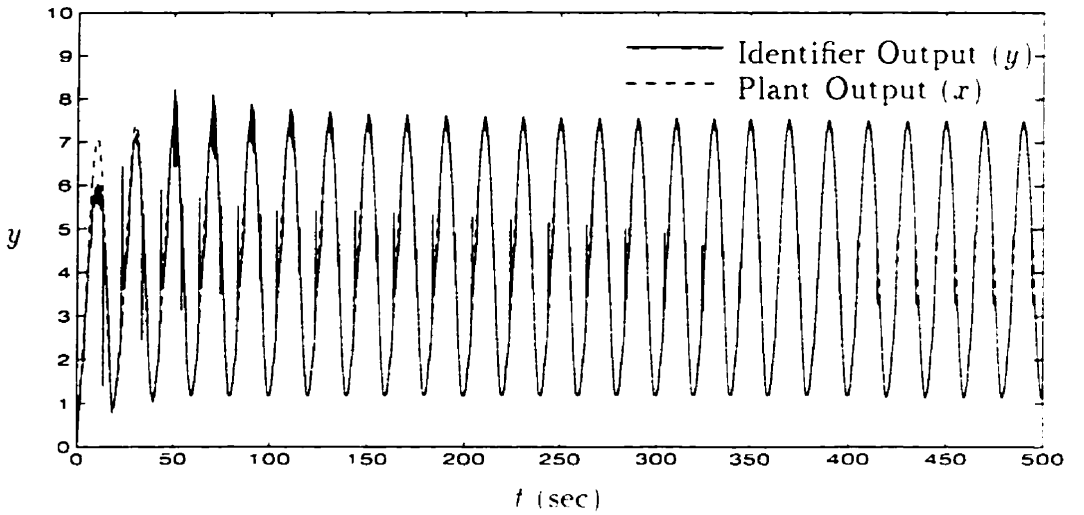
The initial values of the elements of the parameter vector, $\overline{\mathbf{Y}}(0)$, are assigned random numbers uniformly distributed in $(-1, 1)$. The gain matrix, $\mathbf{H}$, is defined as a diagonal matrix with all diagonal elements equal to 175. The bound of the parameter vector, $M_{\overline{\mathbf{Y}}}$, is set to 10^3. Let $\alpha \triangleq 10$, and we assume the system to be identified is initially at rest, i.e., $x(0) = 0$. For input $u = \sin(0.1\pi t)$, the identification processes for x is shown in fig.10.2, where training stops at t=400 sec. It is seen that the training process converges quite well, and after training ends the identifier predicts the system state, x, satisfactorily.

10.4.2 Example 2 – Duffing Forced-Oscillation System

We now demonstrate another application. Consider the following Duffing forced-oscillation System [184].

$$\begin{cases} \dot{x}_1 &= x_2 , \\ \dot{x}_2 &= -x_1^3 - 0.1x_2 + 12\cos(t) + u(t) . \end{cases} \tag{10.93}$$

Our objective is to identify the state variables, x_1 and x_2, by using the RDFLS identification algorithm. As in previous simulation examples, the system dynamic model is treated strictly as a black box.



(Adaptation Terminates at t=400 sec)

Figure 10.2: Identification of x with RDFLS — Example 1

Let

$$\begin{cases} \mathbf{w}_1 \triangleq \{w_{11}, w_{12}, w_{13}\}^T \triangleq \{y_1, x_2, u\}^T . \\ \mathbf{w}_2 \triangleq \{w_{21}, w_{22}, w_{23}\}^T \triangleq \{x_1, y_2, u\}^T . \end{cases} \tag{10.94}$$

Two RDFLS identifiers, D_{x_1} and D_{x_2}, are used to identify the two state variables, x_1 and x_2, respectively, i.e.,

$$\begin{cases} y_1 = D_{x_1}(\mathbf{w}_1') . \\ y_2 = D_{x_2}(\mathbf{w}_2') . \end{cases} \tag{10.95}$$

where y_k represents estimate of the state variable, x_k, and $\mathbf{w}_k'$ is the filtered value of $\mathbf{w}_k$, $k = 1, 2$. Both identifiers, D_{x_1} and D_{x_2}, have three inputs and one output. The identification errors, ϵ_1 and ϵ_2, are defined as

$$\begin{cases} \epsilon_1 \triangleq y_1 - x_1 . \\ \epsilon_2 \triangleq y_2 - x_2 . \end{cases} \tag{10.96}$$

A RDFLS Identifier for x_1 : D_{x_1}

In all the universes of discourse of input variables, five primary fuzzy sets, $A_{p j_p}$, $p = 1, 2, 3$, $j_p = 1, \ldots, 5$, are defined. Gaussian type membership functions are used

for all the primary fuzzy sets, which are the same as those defined in eqs.(8.74-8.76). The prefilters for w_{11}, w_{12} and w_{13} are defined as

$$\begin{cases} w'_{11} = f_1(w_{11}) \triangleq w_{11} \cdot \frac{1}{2} , \\ w'_{12} = f_2(w_{12}) \triangleq w_{12} \cdot \frac{1}{3} , \\ w'_{13} = f_3(w_{13}) \triangleq w_{13} \cdot 2 . \end{cases} \tag{10.97}$$

The output of D_{z_1}, y_1, is specified by the RDFLS of eq.(10.1), i.e.,

$$\dot{y}_1 = -a_1 y_1 - \mathbf{\Theta}_1^T(\mathbf{w}'_1)\overline{\mathbf{Y}}_1 . \tag{10.98}$$

where

$$\begin{cases} \mathbf{\Theta}_1(\mathbf{w}'_1) \triangleq \{\theta_{11}(\mathbf{w}'_1), \theta_{12}(\mathbf{w}'_1), \ldots, \theta_{1(125)}(\mathbf{w}'_1)\}^T , \\ \overline{\mathbf{Y}}_1 \triangleq \{\overline{y}_{11}, \overline{y}_{12}, \ldots, \overline{y}_{1(125)}\}^T . \end{cases} \tag{10.99}$$

and

$$\theta_{1i}(\mathbf{w}'_1) = \frac{\prod_{p=1}^{3} \mu_{A_p^i}(w'_{1p})}{\sum_{i=1}^{125} \prod_{p=1}^{3} \mu_{A_p^i}(w'_{1p})} , \tag{10.100}$$

and for $i = 1, \ldots, 125$, $p = 1, 2, 3$,

$$A_p^i \in \{A_{pj_p} : j_p = 1, \ldots, 5\} . \tag{10.101}$$

The free parameter vector, $\overline{\mathbf{Y}}_1$, is adjusted with the adaptive law, eq.(10.39). We approximate the derivative by a difference to give

$$\overline{\mathbf{Y}}_1(kT + T) = \overline{\mathbf{Y}}_1(kT) - T[\mathbf{H}_1\mathbf{\Theta}_1(\mathbf{w}'_1(kT))e_1(kT) + S_1\beta_1\mathbf{H}_1\overline{\mathbf{Y}}_1(kT)] \tag{10.102}$$

where T is the time incremental step and is set at 0.1 sec. The initial values of the elements of the parameter vector, $\overline{\mathbf{Y}}_1(0)$, are assigned random numbers uniformly distributed in $(-1, 1)$. The gain matrix, $\mathbf{H}_1$, is defined as a diagonal matrix with all diagonal elements set to 100. The bound of the parameter vector, $M_{\overline{\mathbf{Y}}_1}$, is set to 10^3. Let $a_1 \triangleq 10$, and we assume the system to be identified is initially at rest.

For the input $u = \sin(t)$, the system trajectory in state space is shown in fig.10.3, and the identification processes for x_1 is shown in fig.10.4(a), where training ends at $t = 50$ sec. It is seen that the convergence speed and accuracy of the training process are quite satisfactory, and after the end of training, the identifier predicts the system state, x_1, very well.

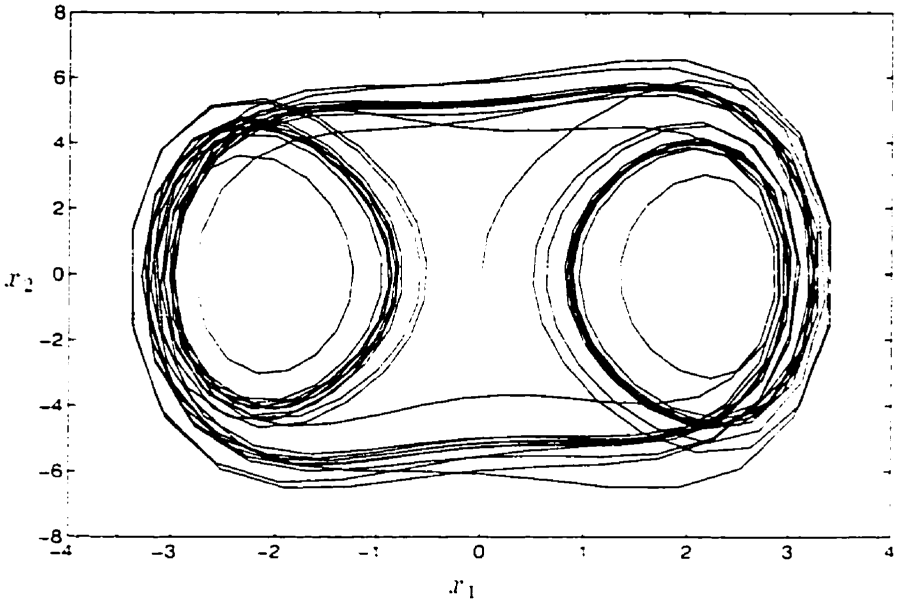


Figure 10.3: State Space Trajectory — Example 2

B RDFLS Identifier for x_2 : D_{x_2}

In all the universes of discourse of input variables, five primary fuzzy sets, B_{j_p}, $p = 1, 2, 3$, $j_p = 1, \ldots, 5$, are defined. Gaussian type membership functions are used for all the primary fuzzy sets, which are the same as those defined in eqs.(8.74-8.76). The prefilters for w_{21}, w_{22} and w_{23} are the same as those defined for D_{x_1}, i.e.,

$$\begin{cases} w'_{21} = f_1(w_{21}) \triangleq w_{21} \cdot \frac{1}{2} , \\ w'_{22} = f_2(w_{22}) \triangleq w_{22} \cdot \frac{1}{3} , \\ w'_{23} = f_3(w_{23}) \triangleq w_{23} \cdot 2 , \end{cases} \tag{10.103}$$

The output of D_{x_2}, y_2, is defined with the RDFLS of eq.(10.1), i.e.,

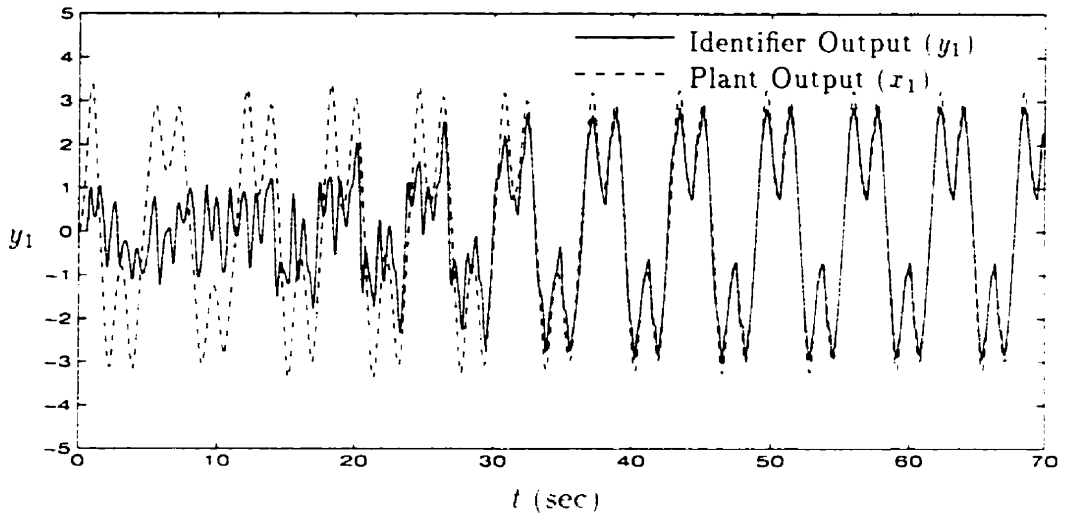
$$\dot{y}_2 = -\alpha_2 y_2 + \mathbf{\Theta}_2^T(\mathbf{w}_2')\overline{\mathbf{Y}}_2 \ . \tag{10.104}$$

where $\mathbf{\Theta}_2(\mathbf{w}_2)$ and $\overline{\mathbf{Y}}_2$ are the same as those defined in eqs.(10.99-10.101). The parameter vector, $\overline{\mathbf{Y}}_2$, is adaptively adjusted with the training law, eq.(10.39), and we approximate the derivative by a difference so that

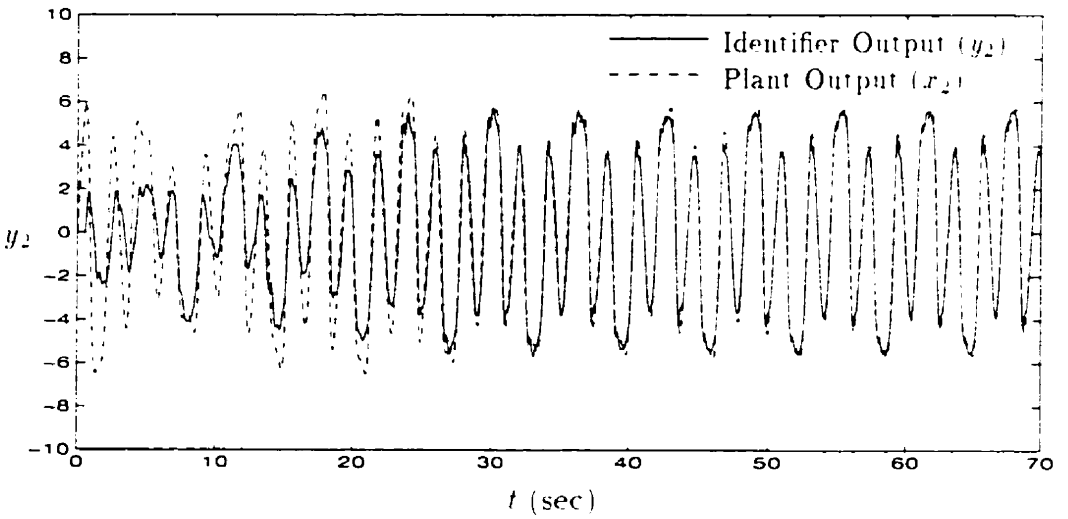
$$\overline{\mathbf{Y}}_2(kT+T) = \overline{\mathbf{Y}}_2(kT) - T[\mathbf{H}_2\mathbf{\Theta}_2(\mathbf{w}_2'(kT))\epsilon_2(kT) + S_2\beta_2\mathbf{H}_2\overline{\mathbf{Y}}_2(kT)] \ . \tag{10.105}$$

We set the time increment, T, to 0.1 sec. and let $\alpha_2 \hat{=} 10$. The initial values of the elements of the parameter vector, $\overline{\mathbf{Y}}_2(0)$, are assigned random numbers uniformly distributed in $(-1, 1)$. The gain matrix, $\mathbf{H}_2$, is defined as a diagonal matrix with all diagonal elements set to 100. The bound of the parameter vector, $M_{\overline{\mathbf{Y}}_2}$, is set to 10^3.

With the system to be identified initially at rest, and for input $u = \sin(t)$, the identification processes for x_2 is shown in fig.10.4(b), where training ends at $t = 50$ sec. Again, the convergence speed and accuracy of the training process are quite satisfactory, and after the end of training, the identifier predicts system state, x_2, very well.



(a) Identification of State Variable x_1



(b) Identification of State Variable x_2

(Adaptation Terminates at t=50 sec)

Figure 10.4: Identification of State Variables with RDFLS — Example 2

10.5 Concluding Remarks

- The RDFLS has been shown to possess the desired universal approximation capability, just as the static FLS and DFLS.

- A RDFLS identification algorithm was developed and described here, which can be used to identify a large class of nonlinear dynamic systems, with identifier stability properties specified in Theorem 10.2, and applications demonstrated in simulation examples.

- One of the features of a well trained RDFLS identifier is that it does not use the plant signal being estimated after the adaptation process ends, but continues to work well, as illustrated in the simulation examples. Therefore RDFLS identifiers may be very useful in the situation where a plant signal to be estimated is difficult or expensive to obtain. This is still open to further investigation.

Chapter 11

Conclusions and Suggestions for Future Work

11.1 Conclusions

The main objective of this work was to develop effective fuzzy logic approaches for identification and control of certain types of ill-defined, nonlinear dynamical systems. This objective was successfully achieved in this research through investigation of a broad range of topics including fuzzy logic systems, nonlinear system identification and adaptive control. The major accomplishments and observations are summarized as follows:

1. In appendix A, a dynamic model of a flexible single link robotic manipulator was developed independently in this work. Although presented as an appendix, it is an integral portion of this research because it provided not only an important vehicle for simulation and experimental demonstration of the theory, but the derivation process itself also provided much insight into the mathematical model and provided clues for the formulation of the research plan in the early stages of this work.

The approach illustrated key assumptions used to obtain a linear dynamic model of the system (not yet considering the highly nonlinear joint friction) which is used in controller design for flexible link manipulators and can be solved by standard mathematical approaches, such as modal analysis and others. It is known that some of the effects generally ignored can be crucial for successful control in certain situations. This inspired us to investigate identification and control approaches which do not explicitly rely on accurate and tractable system mathematical models.

Observing the fact that a human expert is able to control many very complicated systems only through accumulation of expertise, without necessarily knowing their mathematical models, it was decided to investigate the fuzzy logic approach for identification and control of ill-defined, nonlinear dynamic systems, because this approach provided by far the most effective theoretical framework to systematically incorporate human linguistic information and express it in mathematical terms.

2. Two on-line quantitative measures, the CI and FDI, were introduced in section 5.1 for IF-THEN rule performance in fuzzy logic systems, which can indicate certain problems in a fuzzy rule base, such as incompleteness or uneven share of usage of individual rules. With this information, system operators can be warned of these problems and take corresponding measures in time, so that the resulting fuzzy logic systems are expected to be more reliable. In addition, the implementation of the indices produces little extra computational burden. Numerical examples were presented to demonstrate applications of these indices.

3. A statistical confidence measure, the confidence interval, was introduced in section 5.2 for the fuzzy logic systems used in functional approximation prob-

lems, whereby the approximation accuracy of both overall FLS output and prediction of individual IF-THEN rules can be measured and analysed statistically. Numerical examples were also presented to demonstrate applications.

4. Simplification was made of fuzzy inference computation in section 5.3 for an important type of FLS, and significant increase in calculation speed of fuzzy inference was observed, which is especially important for on-line identification and control applications. A closed form expression for the bounds of the simplification errors was derived, and the values of the error bounds were tabulated for applications of potential interest.

5. A complete procedure was presented in chapter 6 for formulating expert knowledge based fuzzy logic controllers, or empirical FLC. It was explicitly demonstrated that human knowledge in linguistic form could be systematically incorporated into fuzzy logic systems and expressed in mathematical form.

   Also it was illustrated, both experimentally and numerically, that the fuzzy logic controllers designed solely on the basis of expert knowledge were capable of performing various control tasks satisfactorily, without knowledge of the system mathematical models.

6. Via experiments, the empirical fuzzy logic control approach was compared to PD control approach in section 6.4. It was observed that although in some simple control tasks the results of the fuzzy logic approach were only slightly better than that of the PD approach, for more complicated situations human expertise could be of much more significance for successful control of these complex systems, where the simple PID control approach might not work at all. Also, the design procedures of FLC and PID control are fundamentally different. The latter is basically a trial and error approach, while the FLC approach provides a systematic framework to explicitly incorporate human

expertise in the form of IF-THEN rules, which can be rationally set up and revised. Furthermore, the fuzzy logic control approach has the potential to be further developed into an effective approach for dealing with highly nonlinear and ill-defined systems where the overall system stability properties and performance criteria can be theoretically analysed and adaptively adjusted. On the other hand the PID approach intrinsically lacks means of formal synthesis for stability analysis in these situations, which often limits its usefulness.

7. An optimal fuzzy logic system training scheme was proposed in chapter 7, which combined the backpropagation training algorithm with the least squares estimation technique. The resulting optimal LSE-BP training scheme avoids the weaknesses of both the LSE and BP approaches, such as the rank requirement in LSE and the slow convergence rate and the possibility of being trapped in local minima for BP approach. Meanwhile, it combines the strong points of both approaches, such as the wide applicability of BP and quick convergence rate and global optimality of LSE. Also, comparisons were made between LSE-BP and BP approaches, and the superiority of the LSE-BP approach was demonstrated.

8. In section 8.2, a novel fuzzy logic system structure which is characterized by its dynamic feature, was proposed, and its universal approximation property proved. This new dynamic fuzzy logic system (DFLS) is believed to be more naturally integrated into dynamic systems and to make better use of intrinsic dynamics than does the conventional static version.

9. A DFLS based stable adaptive identification algorithm was developed in section 8.3 via a Lyapunov synthesis approach, which enables DFLS identifiers to identify a large class of nonlinear dynamic systems reliably.

In section 8.4, the DFLS based identification algorithm was applied to identify a variety of nonlinear systems. The design procedures for these DFLS identifiers were demonstrated in detail, and in all applications satisfactory results were observed, which illustrates the effectiveness of the DFLS identifier in dealing with complex, nonlinear dynamic systems.

DFLS identifiers were also compared with static FLS identifiers (subsection 8.4.2) via a simulation example, and better performance of the DFLS identifier was observed.

10. In most of the DFLS identification applications it was assumed that there was no available *a priori* human expertise, which poses more stringent conditions on DFLS identifiers. Satisfactory results could nevertheless still be achieved.

    If there is expert knowledge available *a priori* in form of IF-THEN rules, even of a very limited nature, it can be incorporated into DFLS identifiers with resulting significantly improved system performance, as observed in the simulation experiment in section 8.4.4.

11. In chapter 9, a novel DFLS based indirect adaptive control scheme was developed via a Lyapunov synthesis approach for a class of nonlinear systems, eq.(9.1),

$$x^{(N)} = f(\mathbf{x}) + g(\mathbf{x})u \ .$$

    The closed loop system performance and stability properties were analysed and summarized in the form of theorems and corollaries, which theoretically support the observed results that this control scheme enabled us to achieve reliable performance even under rather stringent conditions.

12. Two approaches were presented in section 9.5 to estimate the unknown function, $g(\mathbf{x})$, in eq.(9.1). One is based on a traditional self-tuning scheme, the

other is a FLS approach.

Generally speaking, if there is no expert knowledge about g available the former approach may be more appropriate, since it is more straightforward and computationally simpler. If there is such expert knowledge available the FLS approach may be better, because it can explicitly incorporate this knowledge and produce better initial values.

13. The DFLS based adaptive control algorithm was applied to control a variety of nonlinear systems (sections 9.9–9.10). The design procedures of DFLS controllers were illustrated in detail. In all the situations satisfactory results were observed, and the effectiveness of DFLS controllers in dealing with complex, nonlinear systems was demonstrated.

    Notably, experimental results in section 9.10 were very satisfactory when the DFLS adaptive controller was applied to control the moving trajectory of the tip of a flexible beam manipulator, which is a real mechanical system for identification and control. This further demonstrated the effectiveness of the DFLS adaptive control approach in dealing with nonlinear, ill-defined systems, and revealed its potential usefulness for practical applications.

14. In the DFLS based adaptive control algorithm a measurable state variable, x_N, was identified rather than identifying the unknown function, $f(\mathbf{x})$, as in the case of the static FLS based control algorithm. It is easier to acquire knowledge about the dynamic behavior of a measurable state variable than to acquire knowledge about the unknown function, $f(\mathbf{x})$, which is not directly accessible to us.

    Also, in the DFLS based control algorithm, the system control input, u, is explicitly included as an input of the DFLS, where any expert knowledge about the relationship between the system input, u, and the system state, x_N,

can be explicitly incorporated. This is in contrast to the situation of the static FLS based control algorithm, where u is not explicitly included as an input of the FLS.

15. In chapter 10, a recurrent DFLS, RDFLS, was introduced and its universal approximation property proved. A RDFLS based stable adaptive identification algorithm was further developed and its stability properties investigated theoretically.

16. The application of the RDFLS identification algorithm to nonlinear systems was demonstrated in section 10.4 via simulation examples, and satisfactory results were observed, which indicated the potential usefulness of the RDFLS in the identification of nonlinear systems.

    For a well trained RDFLS identifier one of its properties is that, it does not use the plant signal being estimated after the adaptation process ends, but it still functions well, as illustrated in the simulation examples. Therefore, RDFLS identifiers might be useful in the situation where a plant signal to be estimated is difficult or expensive to obtain.

11.2 Summary of Major Contributions

1. A novel dynamic fuzzy logic system structure, the DFLS, was proposed and its universal approximation property proved.

2. A novel DFLS based adaptive identification algorithm was developed and its stability properties theoretically analysed. Its application to various nonlinear, ill-defined dynamic systems was demonstrated.

3. A novel DFLS based indirect adaptive control algorithm for a class of nonlinear systems in companion form was developed and its stability properties theo-

retically analysed. Its applications to various nonlinear, ill-defined dynamic systems was demonstrated.

4. Two approaches, a self-tuning and a fuzzy logic approach, were presented to estimate the unknown function, g, in the DFLS adaptive control algorithm; each has its respective advantages in different situations.

5. A recurrent DFLS structure, the RDFLS, was introduced and its universal approximation property proved. A RDFLS based adaptive identification algorithm was also developed and its stability properties theoretically investigated. Its application to various nonlinear dynamic systems was demonstrated.

6. An optimal fuzzy logic system training scheme, the LSE-BP, which combines the advantages of both LSE and BP approaches while avoiding their respective weaknesses, was proposed.

7. We presented a complete developmental procedure for designing a human expert knowledge based fuzzy logic controller, and demonstrated its properties and applications both experimentally and numerically.

8. We simplified fuzzy inference calculations for an important type of fuzzy logic system and derived closed form expressions for the bounds of simplification errors, and tabulated the values of the bounds of errors for potential practical applications.

9. A statistical confidence measure, the confidence interval for fuzzy logic systems was introduced and used in function approximation problems, where the approximation accuracy of both overall FLS output and performance of individual IF-THEN rules can be statistically measured and analysed.

10. We introduced two simple on-line quantitative measures, the CI and FDI, for IF-THEN rule performance in fuzzy logic systems, which resulted in increased reliability of the corresponding fuzzy logic systems.

11. We developed a dynamic model for a flexible single link robotic manipulator, and explicitly indicated assumptions needed to obtain the widely used, mathematically tractable linear model.

12. We compared, via experiments, the empirical fuzzy logic control approach with traditional PD control approach.

13. We compared the results of DFLS identifiers with those of static FLS identifiers.

14. We demonstrated how to incorporate *a priori* expert knowledge in DFLS identifier design and improve system performance.

15. We experimentally implemented the DFLS adaptive control algorithm and successfully controlled the moving trajectory of the tip of the flexible link of a real mechanical manipulator.

16. Empirical fuzzy logic controllers for vibration and position control of the flexible beam of the mechanical manipulator were experimentally implemented.

17. A mechanical single link robotic manipulator, which could be controlled automatically by computer and/or manually by a human operator through a joystick, was developed. It served as a realistic experimental testbed for theoretical developments on identification and control.

18. We developed relevant software packages in the course of this research. These included communication software between controllers, sensors and actuators, and graphics software for real-time on screen display of input and output

samples of both controllers and the plant. The latter was essential for acquiring knowledge of the system dynamic behavior.

11.3 Suggestions for Future Work

In the author's opinion, the following studies are necessary to refine the current research.

- Investigate the effects of system and measurement noises on system performance, and develop measures to diminish their possible detrimental effects.
- Develop explicit criteria for the conditions of *adequate training*, which may include persistent input excitation issues, the choices of fuzzy logic system parameters, and so on.
- Develop RDFLS based adaptive control schemes, and investigate their stability properties.
- Develop analytical means for determining off-line parameters which would complement the current expert knowledge and trial and error based design approaches such that an optimal set of off-line parameters can be obtained.
- Investigate the effects of sampling period on system stability properties.
- Further investigate the quantitative measures, CI and FDI, and the statistical confidence measure, confidence interval, to understand and relate specific values of these indices with fuzzy logic system performance.
- Integrate other types of sensors with higher sampling rates in the experimental testbed to replace the current ultrasonic sensor, whose maximal sampling rate is only 50 Hz.

- Develop intelligent hybrid systems which combine the methodologies of fuzzy logic, artificial neural networks, and genetic algorithms.

Bibliography

[1] Akbarzadeh-Totonchi, M.-R., Jamshidi, M. and Vadiee, N. "A Hierarchical Fuzzy Controller Using Line-Curvature Extraction for a Single Link Flexible Arm." *Proceedings of IEEE International Conference on Fuzzy Systems*, Orlando, FL, 1994, pp.524-529.

[2] Alasty, A., "Force Control of Flexible Manipulators," Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa: *Ph.D. Dissertation*, 1997.

[3] Alasty, A. and Vukovich, G., "Force Control of Flexible Manipulators with Discontinuous Contact and Joint Friction." *Proceedings of the Third IFAC Symposium on Nonlinear Control Systems Design*, Tahoe City, CA, June, 1995, pp.590-595.

[4] Anderson, J. and Rosenfeld, E. *Neurocomputing: Foundations of Research*, Cambridge: The MIT Press, 1989.

[5] Antsaklis, P. (ed.). "Special Issue on Neural Networks in Control Systems." *IEEE Control Systems Magazine*, Vol.10, No.3, April 1990, pp.3-87.

[6] Antsaklis, P., "Neural Networks for Control Systems," *IEEE Transactions on Neural Networks*, Volume 1, No. 2, June 1990, pp. 242-244.

[7] Åström, K.J., and Wittenmank, B., *Adaptive Control*. Addison-Wesley Publishing Company, 1989.

[8] Banks, S.P., *Mathematical Theories of Nonlinear Systems*. London: Pretice-Hall International, 1988.

[9] Barbieri, E and Özguner, Ü., "Unconstrained and Constrained Mode Expansions for a Flexible Slewing Link." *Journal of Dynamic Systems, Measurement and Control*, Volume 110, No.4, December 1988, pp.416-421.

[10] Bartolini, G., Casalino, G., Davoli, F., Mastrella, M., Minciard, R. and Morten, E., "Development of Performance Adaptive Fuzzy Controllers with Application to Continuous Casting Plants." In *Industrial Applications of Fuzzy Control*, Edited by Sugeno, M., Elsevier Science Publishers B.V., 1985, pp. 73-86.

[11] Bartos, F.J., "Fuzzy Logic Reaches Adulthood." *Control Engineering*, Vol.43, No.10, July 1996, pp.50-56.

[12] Benham, P.P. and Crawford, R.J., *Mechanics of Engineering Materials*, New York: John Wiley & Sons, Inc., 1987.

[13] Beres, W., Sasiadek, J.Z. and Vukovich, G., "Control and Dynamic Analysis of Multilink Flexible Manipulator." *Proceedings of IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, May 1993, pp.478-483.

[14] Billings, S.A., "Identification of Nonlinear Systems - A Survey." *Proceedings of IEE*, Part D, Vol.127, 1980, pp.272-285.

[15] Birdwell, J.D. and Wang, Y., "Lyapunov Stability Analysis of Systems Using the Fuzzy-PID Controller." *Proceedings of 1994 American Control Conference*, Baltimore, Maryland, 1994, pp.966-970.

[16] Biswas, S.K. and Klafter, R.D., "Dynamic Modelling and Optimal Control of Flexible Robotic Manipulators," *Proceedings of 1988 IEEE International Control Conference on Robotics and Automation*, Philadelphia, Pennsylvania, April 1988, pp. 15–20.

[17] Bonissone, P.P., Badami, V., Chiang, K.H., Khedkar, P.S., Marcelle, K.W. and Schutten, M.J. "Industrial Applications of Fuzzy Logic at General Electric," *Proceedings of IEEE*, Vol.83, No.3, March 1995, pp.450–465.

[18] Book, W.J., *Modelling, Design and Control of Flexible Manipulator Arms*, Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Mechanical Engineering, April 1974.

[19] Book, W.J., Maizza-Neto, O. and Whitney, D.E., "Feedback Control of Two Beam, Two Joint Systems with Distributed Flexibility," ASME *Journal of Dynamic Systems, Measurement, and Control*, Volume 97, No. 4, December 1975.

[20] Book, W.J. and Majette, M., "Controller Design for Flexible, Distributed Parameter Mechanical Arms Via Combined State Space and Frequency Domain Techniques" *ASME Journal of Dynamic Systems, Measurement and Control*, Volume 105, March 1983, pp.245–254.

[21] Braae, M. and Rutherford, D.A., "Fuzzy Relations in a Control Setting," *Kybernetes*, Volume 7, No. 3, 1978, pp.185–188.

[22] Braae, M. and Rutherford, D.A., "Theoretical and Linguistic Aspects of the Fuzzy Logic Controller," *Automatica*, Vol.15, No.5, 1979.

[23] Brady, G.S., and Clauser, H.R., *Materials Handbook*, 13th Ed., McGraw-Hill, Inc., 1991.

[24] Bristol, E.H., "The Exact Pattern Recognition Adaptive Controllers, A User-Oriented Commercial Success", in Narendra, K.S., ed. *Adaptive and Learning Systems*, New York: Plenum Press, 1986.

[25] Burr, D.J., "Experiments on Neural Net Recognition of Spoken and Written Text." *IEEE Trans. Acoust., speech, Signal Processing*, Vol.36, No.7, July 1988, pp.1162-1168.

[26] Cannon, R.H., Jr., and Schmitz, E., "Initial Experiments on the End-Point Control of a Flexible One-Link Robot." *The International Journal of Robotic Research*, Volume 3, No. 3, Fall 1984, pp. 62-75.

[27] Chang, S. and Zadeh, L.A., "On Fuzzy Mapping and Control." *IEEE Transactions on Systems, Man, and Cybernetics*, Volume SCM-2, 1972, pp.30-34.

[28] Chen, C.C. and Chen, W.-C., "Fuzzy Controller Design by Using Neural Network Techniques." *IEEE Transactions on Fuzzy Systems*, Vol.2, No.3, Aug. 1994, pp.235-244.

[29] Chen, T. and Chen, H., "Approximation Capability to Functions of Several Variables, Nonlinear Functionals, and Operators by Radial Basis Function Neural Networks." *IEEE Transactions on Neural Networks*, Vol. 6, No. 4, July 1995, pp.904-910.

[30] Chen, T. and Chen, H., "Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and its Application to Dynamic Systems." *IEEE Transactions on Neural Networks*, Vol. 6, No. 4, July 1995, pp.911-917.

[31] Chen, Y.Y., "The Global Analysis of Fuzzy Dynamic Systems." *Ph.D. Dissertation*, University of California, Berkeley, CA, 1989.

[32] Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function." *Math. Control Signal Systems*, 2, 1989, pp.303–314.

[33] Daugherity, W.C., Rathakrishnan, B. and Yen, J., "Performance Evaluation of a Self-Tuning Fuzzy Controller." *Proceedings of IEEE International Conference on Fuzzy Systems*, San Diego, California, March 1992, pp. 389–397.

[34] Davis, J.H. and Hirschorn, R.M., "Tracking Control of a Flexible Robot Link." *IEEE Transactions on Automatic Control*, Volume 33, No. 3, March 1988, pp. 238–248.

[35] Davis, L., (ed.) *Genetic Algorithms and Simulated Annealing*, Los Altos, CA: Kaufmann Publishers, 1987.

[36] de wit, C.C., Olsson, H., Aström, K.J., and Lischinsky, P., "Dynamic Friction Models and Control Design." *Proceedings of the American Control Conference*, San Francisco, CA, June, 1993, pp.1920–1926.

[37] Donne, J.D. and Özgüner, Ü., "Neural Control of a Flexible-Link Manipulator." *Proceedings of IEEE International Conference on Neural Networks*, Orlando, FL, 1994, pp.2327–2332.

[38] Franke, D., "Robust Variable Structure Control of Uncertain Distributed Parameter systems." *IFAC Symposium on the Control of Distributed Parameter Systems*, Los Angeles, 1986, pp.263–268.

[39] Freeman, J.A. and Skapura, D.M., *Neural Networks - Algorithms, Applications and Programming Techniques*, Addison Wesley, 1991.

[40] Funahashi, K.I., "On the approximate Realization of Continuous Mappings by Neural Networks." *Neural Networks*, 2, 1989, pp.183–192.

[41] Goodwin, G.C., and Payne, R.L., *Dynamic System Identification: Experiment Design and Data Analysis*, New York, San Francisco, London: Academic Press, inc. 1977.

[42] Goodwin, G.C., and Mayne, D.Q., "A Parameter Estimation Perspective of Continuous Time Model Reference Adaptive Control," *Automatica*, Volume 23, 1987, pp. 57–70.

[43] Gorman, R.P. and Sejnowski, T.J., "Learned Classification of Sonar Targets Using a Massively Parallel Network," *IEEE Trans. Acoust., speech, Signal Processing*, Vol.36, No.7, July 1988, pp.1135–1140.

[44] Guez, A. and Selinsky, J., "A Trainable Neuromorphic Controller," *Journal of Robotic Systems*, Aug. 1988.

[45] Guo, P., Ni, X.Z. and Zheng, J., "Polymer Extrusion Production Control Using Active Recognition and Adaptive Control Systems," *Proceedings of Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, March 1993, pp.779–784.

[46] Haber, R. and Unbehauen, H., "Structure Identification of Nonlinear Dynamic Systems – A Survey on Input/Output Approaches," *Automatica*, Vol.26, 1990, pp.651–677.

[47] Hastings, G.G. and Book, W.J., "Experiments in Optimal Control of a Flexible Arm," *Proceedings of the 1985 American Control Conference*, Boston, MA, June 1985, pp. 728–729.

[48] Hastings, G.G. and Book, W.J., "Verification of a Linear Dynamic Model for Flexible Robotic Manipulators," *Proceedings of 1986 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 1986, pp.1024–1029.

[49] Hopfield, J.. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of Sciences*, U.S.A., Vol.79, April 1982, pp.2554-2558.

[50] Hopfield, J.. "Neurons with Graded Response have Collective Computational Properties Like Those of Two-State Neurons." *Proceedings of the National Academy of Sciences*, U.S.A., Vol.81, May 1984, pp.3088-3092.

[51] Hopfield, J. and Tank, D.W.. "Neural Computation of Decisions in Optimization Problems." *Biolog. Cybern.*, Vol.52, 1985, pp.141-152.

[52] Hornik, K., Stinchcombe, M., and White, H.. "Multilayer Feedforward Networks are Universal Approximators." *Neural Networks*, Vol.2, 1989, pp.359-366.

[53] Hwang, Y.R. and Tomizuka, M.. "Fuzzy Smoothing Algorithms for Variable Structure Systems." *IEEE Transactions on Fuzzy Systems*, Vol.2, No.4, Nov. 1994, pp.277-284.

[54] Hu, F.L. and Ulsoy, A.G.. "Dynamic Modeling of Constrained Flexible Robot Arms for Controller Design." *ASME Journal of Dynamic Systems, Measurement, and Control*, Volume 116, March 1994, pp. 56-65.

[55] Humar, J.L.. *Dynamics of Structures*, Englewood Cliffs, NJ: Prentice Hall, Inc., 1990.

[56] Hunt, K.J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P.J.. "Neural Networks for Control Systems - A Survey." *Automatica*, Vol.28, No.6, 1992, pp.1083-1112.

[57] Ioannou, P.A., and Datta, A., "Robust Adaptive Control: Design, Analysis and Robustness Bounds," in *Foundations of Adaptive Control*, edited by Kokotovic, P.V., Berlin: Springer-Verlag, 1991, pp. 71-152.

[58] Isermann, R., Lachmann, K.-H. and Matko, D., *Adaptive Control Systems*, Hertfordshire, Great Britain: Prentice Hall International (UK) Ltd., 1992.

[59] Isidori, A., *Nonlinear control systems*, 3rd ed., Berlin: Springer-Verlag, 1995.

[60] Jamshidi, M., Vadiee, N., and Ross, T.J., *Fuzzy Logic and Control - Software and Hardware Applications*, Englewood Cliffs, N.J.: PTR Pretice Hall, 1993.

[61] Jang, J.-S.R., "Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm," *Proceedings of the 9th National Conference on Artificial Intelligence*, July 1991, pp.762-767.

[62] Jang, J.-S.R., "Self-Learning Fuzzy Controllers Based on Temporal Back Propagation," *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, 1992, pp.714-723.

[63] Jang, J.-S.R., "ANFIS: Adaptive-Network-Based Fuzzy Inference System", "*IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, 1993, pp.665-685.

[64] Johansen, T.A., "Fuzzy Model Based Control: Stability, Robustness, and Performance Issues," *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 3, 1994, pp221-234.

[65] Johansson, R., *System Modeling and Identification*, Englewood Cliffs, NJ: Prentice-Hall, Inc. 1993.

[66] Kailath, T., *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, Inc. 1980.

[67] Kandel, A. and Langholz, G., (ed.) *Fuzzy Control Systems*. Boca Raton, Ann Arbor, London, Tokyo: CRC Press, 1994.

[68] Kang, H. "Stability and Control of Fuzzy Dynamic Systems via Cell-State Transitions in Fuzzy Hypercubes." *IEEE Transactions on Fuzzy Systems*, Vol.1, No.4, Nov. 1993, pp.267-279.

[69] Kang, M.S. and Yang, B., "Discrete Time Noncollocated Control of Flexible Mechanical Systems Using Time Delay." *Journal of Dynamic Systems, Measurement, and Control*, Volume 116, No.2, June 1994, pp. 217-222.

[70] Karakaşoğlu, A., Sudharsanan, S.I. and Sundareshan, M.K., "Neural Network-Based Identification and Adaptive Control of Nonlinear Systems: A Novel Dynamical Network Architecture and Training Policy." Brighton, England: *Proceedings of 30th IEEE Conference on Decision and Control*, Dec. 1991, pp.180-181.

[71] Karakaşoğlu, Subramania, I., A., Sudharsanan, S.I. and Sundareshan, M.K., "Identification and Decentralized Adaptive Control Using Dynamical Neural Networks with Application to Robotic Manipulators." *IEEE Transactions on Neural Networks*, Vol. 4, No. 6, Nov. 1993, pp.919-930.

[72] Kaufmann, A. and Gupta, M.M., *Introduction to Fuzzy Arithmetic*, New York: Van Nostrand, 1985.

[73] Kickert, W.M.J., and Mamdani, E.H., "Analysis of a Fuzzy Logic Controller", *Fuzzy Sets and Systems*, 1(1), 1978, pp.29-44.

[74] Kiszka, J. B., Gupta, M. M. and Nikiforuk, P.N., "Energistic Stability of Fuzzy Dynamic Systems." *IEEE Transactions on Systems, Man, and Cybernetics*, Volume SMC-15, No. 6, November/December 1985, pp. 783-803.

[75] Kiszka, J.B., Kochańska, M.E. and Sliwińska, D.S., "The Influence of Some Parameters on the Accuracy of a Fuzzy Model." In *Industrial Applications of Fuzzy Control*, Edited by Sugeno, M., Amsterdam, North-Holland: Elsevier Science Publishers B.V., 1985, pp.187-230.

[76] Kohonen, T., "Automatic Formation of Topological Maps of Patterns in a Self-organizing System." *Proceedings of 2nd Scandinavian Conference on Image Analysis*, 1981, pp.214-220.

[77] Kohonen, T., "Self-organized Formation of Topologically Correct Feature Maps." *Biological Cybernetics*, 43, 1982, pp.59-69.

[78] Kohonen, T., *Self-Organization and Associative Memory*, New York: Springer-Verlag, 2nd ed., 1988.

[79] Korenberg, M.J., "Parallel Cascade Identification and Kernel Estimation for Nonlinear Systems." *Ann. Biomed. Eng.*, Vol.19, 1991, pp.429-455.

[80] Kosko, B., "Fuzzy Systems as Universal Approximators." *Proceedings of IEEE International Conference on Fuzzy Systems*, San Diego, California, March 1992, pp.1153-1162.

[81] Kosmatopoulos, E.B., Polycarpou, M.M., Christodoulou, M.A., and Ioannou, P.A., "High-Order Neural Networks Structures for Identification of Dynamical Systems." *IEEE Transactions on Neural Networks*, Volume 6, No. 2, March 1995, pp. 422-431.

[82] Krishnan. H. and Vidyasagar. M.. "Control of a Single-Link Flexible Beam Using a Hankel-Norm-Based Order Model." *Proceedings of 1988 IEEE International Control Conference on Robotics and Automation*. Philadelphia. Pennsylvania. April 1988. pp. 9–14.

[83] Ku. C.-C. and Lee. K.Y.. "Diagonal Recurrent Neural Networks for Dynamic Systems Control." *IEEE Transactions on Neural Networks*. Vol. 6. No. 1. Jan. 1995. pp. 144–156.

[84] Kung. C.-C. and Liao. C.-C.. "Fuzzy-Sliding Mode Controller Design for Tracking Control of Nonlinear Systems." *Proceedings of 1994 American Control Conference*. Baltimore. Maryland. 1994. pp.180–184.

[85] Langari. G. and Tomizuka. M.. "Stability of Fuzzy Linguistic Control Systems." *Proceedings of the 29th Conference on Decision and Control*. Honolulu. Hawaii. Dec. 1990. pp.2185–2190.

[86] Larkin. L.I.. "A Fuzzy Logic Controller for Aircraft Flight Control." In *Industrial Applications of Fuzzy Control*. Edited by Sugeno. M.. Amsterdam. North-Holland: Elsevier Science Publishers B.V.. 1985. pp.87–103.

[87] Larsen. P. M.. "Industrial Applications of Fuzzy Logic Control." In *Fuzzy Reasoning and its Applications*. Edited by Mamdani. E.H. and Gaines. B.R.. Academic Press. 1981. pp.335–342.

[88] Lee. C. C.. "Fuzzy Logic in Control Systems. Fuzzy Logic Controller-Part I and II." *IEEE Transactions on Systems. Man. and Cybernetics*. Volume 20. No. 2. March/April 1990. pp. 404–435.

[89] Lee. J.. "On Methods for Improving Performance of PI-Type FLCs." *IEEE Transactions on Fuzzy Systems*. Vol.1. No.4. Nov. 1993. pp.298–301.

[90] Lee, J.X., *Vibration Analysis of Linearly Elastic Non-Conservative Systems Using Integral Equations*. M.Eng. Thesis, Dept. of Mechanical and Aerospace Engineering, Carleton University, Ottawa, 1991.

[91] Lee, J.X., Vukovich, G., and Sasiadek, J.Z., "Fuzzy Control of A Flexible Link Manipulator," *Proceedings of 1994 American Control Conference*, Baltimore, U.S.A., June 1994, pp.568-574.

[92] Lin, C.-L., "Robust Control of Flexible Structures Using Residual Model Filters," *AIAA Journal of Guidance, Control, and Dynamics*, Vol.16, No.5, 1993, pp.973-977.

[93] Lin, L.-C. and Yih, T.-W., "Rigid Model-Based Neural Network Control of Flexible-Link Manipulators," *IEEE Transactions on Robotics and Automation*, Vol.12, No.4, Aug. 1996, pp.595-602.

[94] Ling, C. and Edger, T., "A New Fuzzy Gain Scheduling Algorithm for Process Control," *Proceedings of 1992 Americal Control Conference*, Chicago, IL, 1992, pp.2284-2290.

[95] Linsker, R., "Self-Organization in a Perceptual Network," *Computer*, March 1988, pp.105-117.

[96] Luo, Z., "Direct Strain Feedback Control of Flexible Robot Arms: New Theoretical and Experimental Results," *IEEE Transactions on Automatic Control*, Volume 38, No.11, Nov. 1993, pp.1610-1622.

[97] Maiers, J. and Sherif, Y.S., "Applications of Fuzzy Set Theory," *IEEE Transactions on Systems, Man, and Cybernetics*, Volume SMC-15, No. 1, January/February 1985, pp. 175-189.

[98] Malki, H.A., Li, H., and Chen, G., "New Design and Stability Analysis of Fuzzy Proportional-Derivative Control systems." *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 4, 1994, pp.245-254.

[99] Mamdani, E.H., "Application of Fuzzy Algorithm for Control of Simple Dynamic Plant." *Proc. IEEE*, 121:1585-1588, 1974.

[100] Mamdani, E.H., "Application of Fuzzy Logic to Approximate reasoning." *IEEE Transactions on Computers*, Vol.26, 1977, pp.1182-1191.

[101] Mamdani, E.H., "Advances in the Linguistic Synthesis of Fuzzy Controllers." In *Fuzzy Reasoning and its Applications*, Edited by Mamdani, E.H. and Gaines, B.R., Academic Press, 1981, pp.325-334.

[102] Mamdani, E.H., "Twenty Years of Fuzzy Control : Experiences Gained and Lessons Learnt." *Proceedings of the Second IEEE International Conference on Fuzzy Systems*, San Francisco, California, March 1993, pp.339-344.

[103] Mamdani, E.H. and Assilian, S., "A Case Study on the Application of Fuzzy Set Theory to Automatic Control." *Proceedings of IFAC Stochastic Control Symposium*, Budapest, Hungary, 1974.

[104] Mamdani, E.H. and Assilian, S., "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller." *International Journal of Man, Machine Studies*, Volume 7, 1975, pp. 1-13.

[105] McCulloch, W. and Pitts, W., "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics*, Vol.5, 1943, pp.115-133.

[106] Mehra, R.K., "Nonlinear System Identification - Selected Survey and Recent Trends." *Proceedings of 5th IFAC Symposium on Identification and System Parameter Estimation*, Darmstadt, 1979.

[107] Meirovich, L., *Analytical Methods in Vibrations*, New York: The Macmillan Company, 1967.

[108] Melsa, P.J.W., *Neural Networks: A Concept Overview*, Tellabs Research Center, Mishawaka, IN: Technical Report, TRC-89-08, 1989.

[109] Mendel, J.M., "Fuzzy Logic Systems for Engineering: a Tutorial." *Proceedings of the IEEE - Special Issue on Engineering Applications of Fuzzy Logic*, Vol.83, No.3, March 1995.

[110] Miller, W.T., III, Sutton, R.S. and Werbos, P.J., *Neural Networks for Control*, Cambridge, Massachusetts: London, England: The MIT Press, 1990.

[111] Morgül, Ö. "Dynamic Boundary Control of a Euler-Bernoulli Beam." *IEEE Transactions on Automatic Control*, Volume 37, No. 5, May 1992, pp.639-642.

[112] Müller, B. and Reinhardt, J., *Neural Networks - An Introduction*, Springer-Verlag, 1990.

[113] Murakami, S. and Maeda, M., "Automobile Speed Control System Using a Fuzzy Logic Controller." In *Industrial Applications of Fuzzy Control*, Edited by Sugeno, M., Amsterdam, North-Holland: Elsevier Science Publishers B.V., 1985, pp.105-124.

[114] Narendra, K.S., and Annaswamy, A.M., *Stable Adaptive Systems*, Englewood Cliffs, NJ: Prentice Hall, 1989.

[115] Narendra, K.S. and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks." *IEEE Transaction on Neural Networks*, Volume 1, No. 1, March 1990, pp. 4-26.

[116] Narendra, K.S. and Parthasarathy, K., "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks." *IEEE Transaction on Neural Networks*, Volume 2, No. 2, March 1991, pp.252-262.

[117] Nelson, W.L., Mitra, D. and Boie, R.A., "End-Point Sensing and Load-Adaptive Control of a Flexible Robot Arm." *Proceedings of 24th Conference on Decision and Control*, Ft. Lauderdale, Florida, December 1985, pp. 1410-1415.

[118] Newton, R.T. and Xu, Y., "Real-Time Implementation of Neural Network Learning Control of a Flexible Space Manipulator." *Proceedings of IEEE International Conference on Robotics and Automation*, Atlanta, GA, 1993, pp.135-141.

[119] Nijmeijer, H., *Nonlinear dynamical control systems*, New York: Springer-Verlag, 1990.

[120] Ogata, K., *Modern Control Engineering*, Upper Saddle River, NJ: Prentice Hall, 3rd ed., 1997.

[121] Olafsson, S. and Abu-Mostafa, Y., "The Capacity of Multilevel Threshold Functions." *IEEE Transaction on Pattern Analysis and Machine Inteligence*, 10(2), March 1988, pp.277-281.

[122] Omatu, S. and Seinfeld, J.H., *Distributed Parameter Systems - Theory and Applications*, Oxford: Oxford University Press, 1989.

[123] Palm, R. "Sliding Mode Fuzzy Control." San Diego, CA: *1992 IEEE International Conference on Fuzzy Systems*, March 1992, pp.519-526.

[124] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, Inc., 1991.

[125] Park, J.-H. and Asada, H., "Dynamic Analysis of Noncollocated Flexible Arms and Design of Torque Transmission Mechanisms," *Journal of Dynamic Systems, Measurement, and Control*, Volume 116, No.2, June 1994, pp.201-207.

[126] Pease, B., "What's All This Fuzzy Logic Stuff, Anyhow?" *Electronic Design*, May 13, 1993, pp.77-79.

[127] Pease, B., "What's All This Fuzzy Logic Stuff, Anyhow (Part II)?" *Electronic Design*, Nov. 1, 1993, pp.95-98.

[128] Pedrycz, W., "Construction of Fuzzy Relational Models," in *Cybernetics and Systems Research Volume II*, Trappl, R. ed., Amsterdam: North Holland, 1984, pp.545-549.

[129] Pedrycz, W., *Fuzzy Control and Fuzzy Systems*, 2nd. extended edition, Taunton, Somerset, England: Research Studies Press Ltd., New York-Chichester-Toronto-Brisbane-Singapore: John Wiley & Sons Ins., 1993.

[130] Pedrycz, W., *Fuzzy Sets Engineering*, Boca Raton, Ann Arbor, London, Tokyo: CRC Press, 1995.

[131] Pedrycz, W., "Fuzzy Multimodels," *IEEE Transactions on Fuzzy Systems*, Vol.4, No.2, May 1996, pp.139-148.

[132] Pedrycz, W., (ed.) *Fuzzy Evolutionary computation*, Boston, London, Dordrecht: Kluwer Academic Publishers, 1997.

[133] Polycarpou, M.M., and Ioannou, P.A., "Stable Nonlinear System Identification Using Neural Network Models," in *Neural Networks in Robotics* ed by Bekey, G. and Goldberg, K., Norwell, MA: Kluwer Academic Publishers, 1993, pp.147-164.

[134] Procyk, T. J. and Mamdani, E.H., "A Linguistic Self-Organizing Process Controller," *Automatica*, Vol.15, Pergamon Press Ltd., 1979, pp. 15-30.

[135] Psaltis, D., Sideris, A. and Yamamura, A.A., "A Multilayered Neural Network Controller," *IEEE Control Systems Magazine*, Vol.8, April 1988, pp.17-21.

[136] Qian, W.T. and Ma, C.C.H., "A New Controller Design for a Flexible One-Link Manipulator," *IEEE Transactions on Automatic Control*, Volume 37, No. 1, January 1992, January 1992, pp. 132-137.

[137] Qin, S.T., and Borders, G., "A Multiregion Fuzzy Logic Controller for Nonlinear Process Control," *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 1, 1994, pp.74-81.

[138] Qin, S.Z., Su, H.T. and McAvoy, T.J., "Comparison of Four Neural Net Learning Methods for Dynamic System Identification," *IEEE Transactions on Neural Networks*, Volume 3, No. 1, January 1992, pp. 122-130.

[139] Rao, S.S., *Mechanical Vibrations*, 2nd ed., Reading, Mass.: Addison-Wesley, 1990.

[140] Rauch, H. and Winarske, T., "Neural Networks for Routing Communication traffic," *IEEE Control Systems Magazine*, Vol.8, No.2, April 1988, pp.26-30.

[141] Roach, G.F., *Green's Functions*, Second Edition, Cambridge: Cambridge University Press, 1981.

[142] Rovner. D.M. and Cannon. R.H., Jr., "Experiments Toward On-Line Identification and Control of a Very Flexible One-Link Manipulator." *The International Journal of Robotic Research*. Volume 6. No. 4. Winter 1987. pp.3-19.

[143] Rovner. D.M. and Franklin. G.F., "Experiments In Load-adaptive Control of a Very Flexible One-link Manipulator." *Automatica*. Volume 24. No. 4. 1988. pp.541-548.

[144] Rumelhart. D., Hinton. G.E. and Williams. R.J., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: The MIT Press. 1986.

[145] Sadegh. N., "A Perceptron Network for Functional Identification and Control of Nonlinear Systems." *IEEE Transactions on Neural Networks*. Vol.4. No.6. Nov. 1993. pp.982-988.

[146] Saridis. G.N., "Toward the Realization of Intelligent Controls." *Proc. IEEE*. 1979. pp.1115-1133.

[147] Scharf. E. M. and Mandic. N. J., "The Application of a Fuzzy Controller to the Control of Multi-Degree-of-Freedom Robot Arm." In *Industrial Applications of Fuzzy Control*. Edited by Sugeno. M., Amsterdam. North-Holland: Elsevier Science Publishers B. V., 1985. pp.41-61.

[148] Sejnowski. T.J. and Rosenberg. C.R., "Parallel Networks that Learn to Pronounce English Text." *Complex. Syst.*, Vol.1. 1987. pp.145-168.

[149] Simo. J.C. and Vu-Quoc. L., "On the Dynamics of Flexible Beams Under Large Overall Motions- The Plane Case: Part I and II." *Journal of Applied Mechanics*. Volume 53. Dec.1986. pp. 849-863.

[150] Singh, S.N. and Schy, A.A., "Control of Elastic Robotic Systems by Nonlinear Inversion and Modal Damping." *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol.18, Sept. 1986, pp.180-189.

[151] Slotine, J.-J. E. and Li, W., *Applied Nonlinear Control*. Englewood Cliffs, NJ : Prentice Hall, 1991.

[152] Spooner, J.T. and Passino, K.M., "Stable Adaptive Control Using Fuzzy Systems and Neural Networks." *IEEE Transactions on Fuzzy Systems*, Vol.4, No.3, Aug. 1996, pp.339-359.

[153] Steinberg, M., "Development and Simulation of a F/A 18 Fuzzy Logic Automatic Carrier Landing System." *Proceedings of Second IEEE International Conference on Fuzzy Systems*, San Francisco, California, March 1993, pp.779 784.

[154] Su, C.-Y., and Stepanenko, Y., "Adaptive Control of a Class of Nonlinear Systems with Fuzzy Logic." *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 4, 1994, pp.285 294.

[155] Sudharsanan, S.I. and Sundareshan, M.K., "Training of a Three-Layer Dynamic Recurrent Neural Network for Nonlinear Input-Output Mapping." *Proceedings of 1991 International Joint Conference on Neural Networks*, Seattle, WA, Vol.II July 1991, pp.111-115.

[156] Sugeno, M., Edited *Industrial Applications of Fuzzy Control*. Amsterdam, North-Holland: Elsevier Science Publishers B.V., 1985.

[157] Sugeno, M. and Kang, G.T., "Fuzzy Modelling and Control of Multilayer Incinerator.", *Fuzzy Sets and Systems*, Vol.18, 1986, pp.329-346.

[158] Suzuki. K.. Naka. Y. and Bito. K.. "Fuzzy Multimodel Control of a High-Purity Distillation System." in *Fuzzy Engineering Toward Human Friendly Systems*. Terano. T.. Sugeno. M.. Mukaidono. M. and Shigemasu. K.. ed. Amesterdam: IOS Press. 1992. pp.684-693.

[159] Takagi. T.. and Sugeno. M.. "Fuzzy Identification of Systems and its Applications to Modeling and Control." *IEEE Transactions on Systems. Man. and Cybernetics*. Vol. SMC-15. No. 1. 1985. pp.116-132.

[160] Takahashi. K. and Yamada. I.. "Neural-Network Based Learning Control of Flexible Mechanism with Application to a Single-Link Flexible Arm." *ASME Journal of Dynamic Systems. Measurement. and Control*. Vol.116. Dec. 1994. pp.792-795.

[161] Tanaka. K.. and Sugeno. M.. "Stability Analysis and design of Fuzzy Control Systems." *Fuzzy Sets Syst.*. Vol.45. No.2. 1992. pp.135-156.

[162] Tank. D.W. and Hopfield. J.J.. "Simple 'Neural' Optimization Networks: An A/D converter. Signal Decision Circuit. and a Linear Programming Circuit." *IEEE Transactions on Systems. Man. and Cybernetics*. Vol.CAS-33. No.5. May 1986. pp.533-541.

[163] Terano. T.. Asai. K.. and Sugeno. M.. *Fuzzy Systems Theory and its Applications*. Boston. San Diego. New York. London. Sydney. Tokyo. Toronto: Academic Press. Inc.. 1992.

[164] Tong. R.M.. "A Control Engineering Review of Fuzzy Systems." *Automatica*. Volume 13. 1977. pp. 559-569.

[165] Tong. R.M.. "The Construction and Evaluation of Fuzzy Models." in *Advances to Fuzzy Set Theory and Applications*. Gupta. M.M. ed.. New York: Plenum. 1979.

[166] Tong, R.M., "An Annotated Bibliography of Fuzzy Control," In *Industrial Applications of Fuzzy Control*. Edited by Sugeno, M., Amsterdam, North-Holland: Elsevier Science Publishers B.V., 1985, pp.249-269.

[167] Tzafestas, S.G., *Distributed Parameter Control Systems*. Oxford, New York, Toronto, Sydney, Paris, Frankfurt: Pergamon Press, 1982.

[168] Tzes, A. and Kyriakides, K., "Adaptive Fuzzy Control for Flexible Link Manipulators: a Hybrid Frequency-Time Domain Scheme," San Francisco, CA.: *Proceedings of the Second IEEE International Conference on Fuzzy Systems*, March 1993, pp.122-127.

[169] Umbers, I.G. and King, P.J., "An Analysis of Human Decision-making in Cement Kiln Control and the Implications for Automation," In *Fuzzy Reasoning and its Applications*, Edited by Mamdani, E.H. and Gaines, B.R., Academic Press, 1981, pp.369-381.

[170] Usoro, P.B., Nadira, R. and Mahil, S.S., "A Finite Element/Lagrange Approach to Modelling Lightweight Flexible Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, Volume 108, September 1986, pp.198-205.

[171] Vachtsevanos, G., Farinwata, S.S. and Kang, H., "A Systematic Design Method for Fuzzy Logic Control with Application to Automative Idle Speed Control," *Proceedings of the 31th Conference on Decision and Control*, Tucson, Arizona, Dec. 1993, pp.2547-2548.

[172] Vachtsevanos, G., Farinwata, S.S. and Pirovolou, D.K., "Fuzzy Logic Control of an Automotive Engine," *IEEE Control Systems Magazine*, June 1993, pp.62-68.

[173] Wang, B.H. and Vachtsevanos, G., "Learning Fuzzy Logic Control: An Indirect Control Approach." *Proceedings of IEEE International Conference on Fuzzy Systems*, San Diego, California, March 1992, pp. 297-304.

[174] Wang, H.O., Tanaka, K. and Griffin, M.F., "Parallel Distributed Conpensation of Nonlinear Systems by Takagi-Sugeno Fuzzy Model." *Proc. Fuzz. IEEE/IFES'95*, 1995, pp.531-538.

[175] Wang, H.O., Tanaka, K. and Griffin, M.F., "An Analytical Framework of Fuzzy Modeling and Control of Nonlinear Systems: Stability and Design Issues." *Proceedings of 1995 American Control Conference*, Seattle, WA, 1995, pp.2272-2276.

[176] Wang, H.O., Tanaka, K. and Griffin, M.F., "An Approach to Fuzzy Control of Nonlinear Systems: Stability and Design Issues." *IEEE Transactions on Fuzzy Systems*, Vol.4, No.1, Feb. 1996, pp.14-23.

[177] Wang, L.X., "Fuzzy Systems Are Universal Approximators." San Diego, California: *IEEE International Conference on Fuzzy Systems*, 1992, pp.1163-1170.

[178] Wang, L.X. and Mendel, J.M., "Back-Propagation Fuzzy System as Nonlinear Dynamic System Identifiers." *Proceedings of IEEE International Conference on Fuzzy Systems*, San Diego, California, March 1992, pp. 1409-1418.

[179] Wang, L.X. and Mendel, J.M., "Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least Squares Learning." *IEEE Transactions on Neural Networks*, Volume 3, No. 5, September 1992, pp. 807-814.

[180] Wang, L.X. and Mendel, J.M., "Generating Fuzzy Rules by Learning from Examples." *IEEE Transactions on Systems, Man, and Cybernetics*, Volume 22, No. 6, November/December 1992, pp. 1414-1427.

[181] Wang, L.X. and Mendel, J.M., "Fuzzy Adaptive Filters, with Application to Nonlinear Channel Equaliztion," *IEEE Transaction on Fuzzy Systems*, Vol.1, No.3, 1993.

[182] Wang, L.X., "Stable Adaptive Control of Nonlinear Systems," *IEEE Transactions on Fuzzy Systems*, Volume 1, No. 2, 1993, pp. 146–155.

[183] Wang, L.X., "A Supervisory Controller for Fuzzy Control Systems that Guarantees Stability," *IEEE Transactions on Automatic Control*, Vol.39, No.9, Sept. 1994, pp.1845-1847.

[184] Wang, L.X., *Adaptive Fuzzy Systems and Control*, Englewood Cliffs, NJ: PTR Prentice Hall, 1994.

[185] Werbos, P.J., "Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences," *Ph.D. Thesis*, Harvard University, Committee on Applied Mathematics, 1974.

[186] Werbos, P.J., "Generalization of Backpropagation with Applications to a Recurrent Gas Market model," *Neural Networks*, Vol.1, 1988, pp.339-356.

[187] White, D.A. and Sofge, D.A., (ed.) *Handbook of Intelligent Control - Neural, Fuzzy, and Adaptive Approaches*, New York: Van Nostrand Reinhold, 1992.

[188] Widrow, B., McCool, J. and Medoff, B. "Adaptive Control by Inverse Modeling," Proceedings of the 12th Asilomar Conference on Circuits, Systems and Computers, 1978.

[189] Widrow, B., "Adaptive Inverse Control," *Proceedings of the 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing*, Lund, Sweden, 1986, pp.1-5.

[190] Widrow, B., Winter, R.G. and Baxter, R.A., "Layered Neural Nets for Pattern Recognition." *IEEE Trans. Acoust., speech, Signal Processing*, Vol.36, No.7, July 1988, pp.1109-1118.

[191] Wiener, N., *Nonlinear Problems in Random Theory*. Cambridge, MA: MIT Press, 1958.

[192] Wigren, T., *Recursive Identification Based on the Nonlinear Wiener Model*. Uppsala, Sweden: Acta Universitatis Upsaliensis, 1990.

[193] Xi, F. and Fenton, R.G., "Inverse Dynamic Analysis of Flexible Link Manipulators." *Proceedings of 14th Canadian Congress of Applied Mechanics*, Kinston, Ontario, Canada, May 1993, pp.61-62.

[194] Xu, H.Y. and Vukovich, G., "Robotic Modelling and Control Using a Fuzzy Neural Network." *Proceedings of 1993 IEEE International Conference on Neural Networks*, San Francisco, California, March 1993, pp. 1004-1009.

[195] Xu, H.Y. and Vukovich, G., "A Fuzzy Genetic Algorithm with Effective Search and Optimization." *Proceedings of International Joint Conference on Neural Networks*, Singapore, 1993, pp.2967-2969.

[196] Yager, R.R. and Filev, D.P., *Essentials of Fuzzy Modeling and Control*, New York, Chichester, Brisbane, Toronto, Singapore: John Wiley & Sons, Inc., 1994.

[197] Yager, R.R., Ovchinnikov, S., Tong, R.M., and Nguyen, H.T., Edited *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, John Wiley & Sons, Inc., 1987.

[198] Yager, R.R. and Zadeh, L.A., (ed.) *An Introduction to Fuzzy Logic Applications in Intelligent Systems.* Boston/Dordrecht/London: Kluwer Academic Publishers, 1992.

[199] Yagishita, O., Itoh, O. and Sugeno, M., "Application of Fuzzy Reasoning to the Water Purification Process." In *Industrial Applications of Fuzzy Control.* Edited by Sugeno, M., Amsterdam, North-Holland: Elsevier Science Publishers B.V., 1985, pp.19-40.

[200] Yang, T.C., Yang, J.C.S. and Kudva, P., "Load-Adaptive Control of a Single-Link Flexible Manipulator." *IEEE Transactions on Systems, Man, and Cybernetics,* Volume 22, No. 1, January/February 1992, pp. 85-91.

[201] Yasunobu, S. and Miyamoto, S., "Automatic Train Operation System by Predictive Fuzzy Control." In *Industrial Applications of Fuzzy Control.* Edited by Sugeno, M., Amsterdam, North-Holland: Elsevier Science Publishers B.V., 1985, pp.1-18.

[202] Yigit, A.S., Scott, R.A., and Ulsoy, A.G. "Flexural Motion of a Radially Rotating Beam Attached to a Rigid Body." *Journal of Sound and Vibration,* Volume 121, No.2, 1988, pp.201-210.

[203] Yurkovich, S., Pacheco, F.E. and Tzes, A.P., "On-Line Frequency Domain Information for Control of a Flexible-Link Robot with Varying Payload." *IEEE Transactions on Automatic Control,* Volume 34, No. 12, December 1989, pp. 1300-1304.

[204] Zadeh, L.A., "Fuzzy Sets." *Information and Control,* Volume 8, New York: Academic Press, 1965, pp. 338-353.

[205] Zadeh, L.A., "A Rationale for Fuzzy Control." *ASME Journal of Dynamic Systems, Measurement, and Control,* Volume 94, 1972, pp.3-4.

[206] Zadeh, L.A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Transactions on Systems, Man, and Cybernetics*, Volume SMC-3, 1973, pp. 28–44.

[207] Zadeh, L.A., "The Evolution of Systems Analysis and Control: A Personal Perspective," *IEEE Control Systems Magazine*, June 1996, pp. 95–98.

[208] Zaki, A.S., Mir, H. and ElMaraghy, W.H., "Initial Experiments on the TWO Flexible Manipulator," Kingston, Ontario: *Proceedings of 14th Canadian Congress of Applied Mechanics*, May 1993, pp.107–108.

[209] Zheng, L., "A Practical Guide to Tune of Proportional and Integral (PI) Like Fuzzy Controller," *Proc. 1st IEEE ICFS*, March 1992, pp.663–640.

[210] Zhao, H. and Ma, M., "The Application of Fuzzy and Artificial Intelligence Methods in the Building of a Blast Furnace Melting Process Model," In *Industrial Applications of Fuzzy Control*, Edited by Sugeno, M., Amsterdam, North-Holland: Elsevier Science Publishers B.V., 1985, pp. 241–247.

[211] Zienkiewicz, O.C., *The Finite Element Method*, Third Edition, McGraw-Hill, 1977.

[212] Zimmermann, H.-J., *Fuzzy Set Theory and its Applications*, 3rd ed., Boston/Dordrecht/London: Kluwer Academic Publishers, 1996.

[213] *Logitech 3D Mouse and Head Tracker Technical Reference Manual*, Logitech Inc., Fremont, CA94555, 1992.

[214] *Compumotor-Dynaserv DM & DR Series User Guide*, Compumotor Division, Parker Hannifin Corporation, 5500 Business Park Dr., Rohnert Park, CA94928 USA, 1992.

[215] *Model 5312 Quadrature Encoder Input-PC User's Manual.* Technology 80. Inc., 658 Mendelssohn Ave., Minneapolis, MN55427 USA, 1991.

[216] *RTI-800/815 Hardware Manual* and *RTI-800/815 Software Manual.* Analog Devices, Inc., One Technology Way, Norwood, MA02062-9902 USA, 1990.

Appendix A

Dynamic Model of Flexible Single Link Manipulator

A.1 System Configuration

A mechanical flexible single link robotic manipulator is illustrated in fig.A.1, which has been serving as an experimental testbed for identification and control studies in Robotics Laboratory of Canadian Space Agency. The robotics manipulator of this kind may be schematically shown in fig.A.2.

It is assumed that the link rotates within the horizontal plane and deflects freely in this plane, but is rigid in the vertical plane. The link is viewed as an elastic pinned-free beam with a load M_t at its free end. The rotor of the actuator is considered part of the beam located at the pinned end with mass moment of inertia J_h, on which a torque T_h is exerted. $\Phi(t)$ is the angular position of the rigid mode of the beam, $o(x,t)$ is the actual angular position at x along the beam, and $\alpha(x,t)$ is the angle of deflection due to the structural flexibility. We have

$$o(x,t) = \Phi(t) + \alpha(x,t) \ . \tag{A.1}$$

In fig.A.2, $\vec{i}o\vec{j}$ is the fixed reference frame with origin o at the center of the motor hub, $\vec{a}_i o\vec{a}_j$ is a rotating frame that follows the rigid mode of the beam, and $\vec{s}_i s_j \vec{s}_j$ is

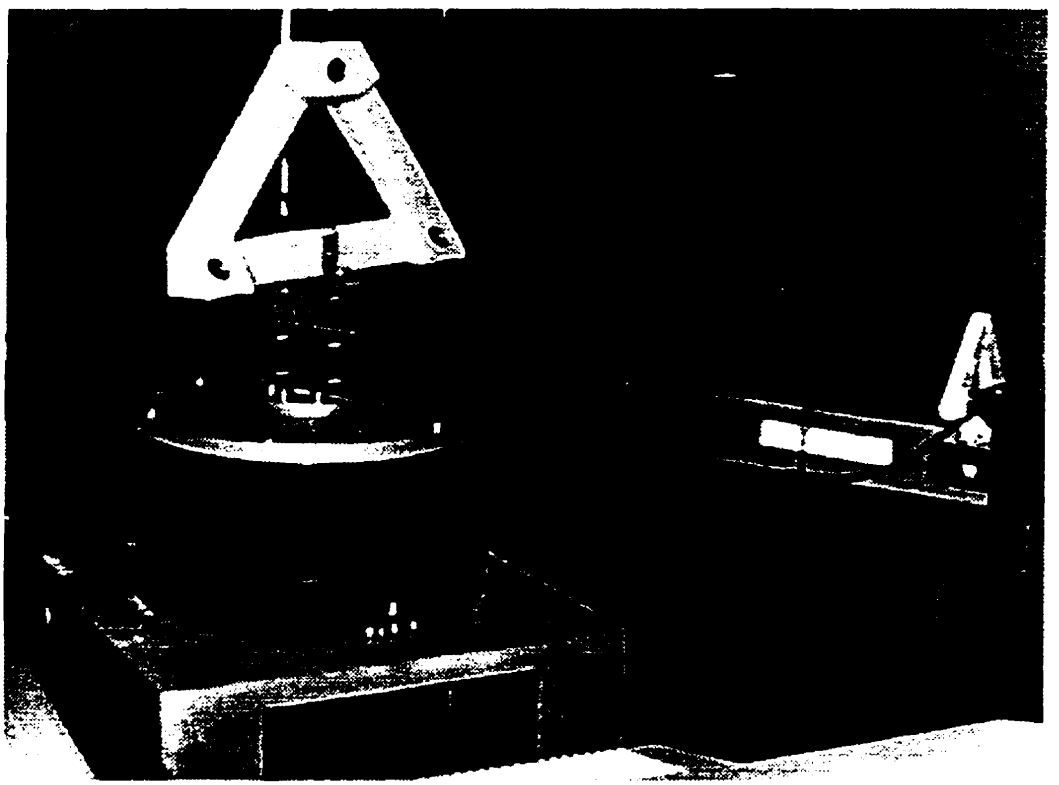


Figure A.1: An Experimental Single Link Robotic Manipulator

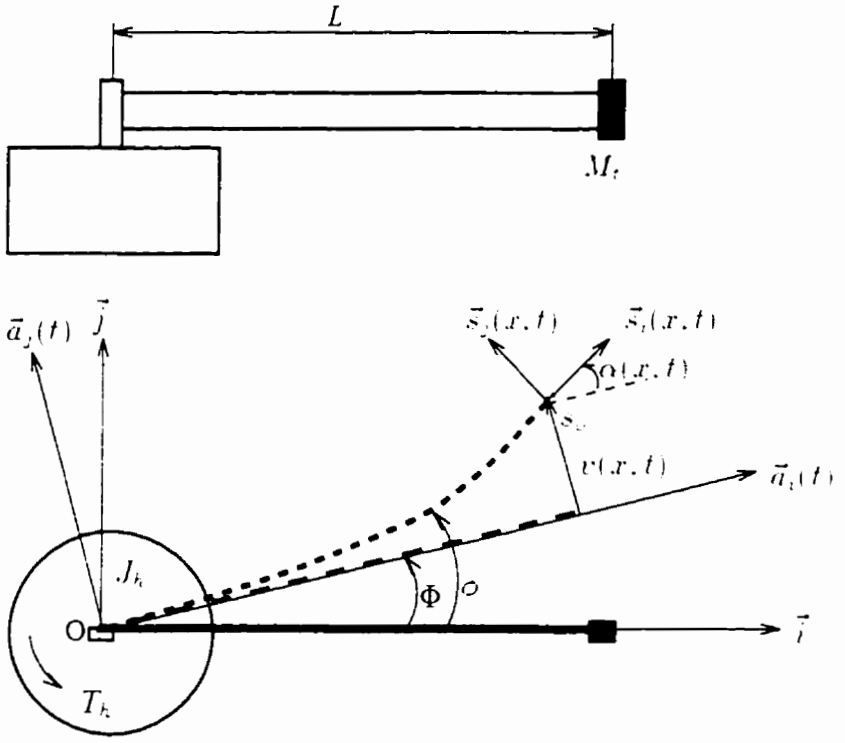


Figure A.2: A Schematic Illustration of Single Link Robotic Manipulator

a moving frame fixed on a cross-section of the beam with $\vec{s}_j$ always contained in the deformed cross-section and $\vec{s}_i$ perpendicular to the cross-section. Since the motion is planar, it has $\vec{k} \equiv \vec{a}_k \equiv \vec{s}_k$. The following relationships will be useful in later development [149].

$$\begin{cases} \dot{\vec{a}}_i(t) &= \dot{\Phi}(t)\, \vec{a}_j(t) \ . \\ \dot{\vec{a}}_j(t) &= -\dot{\Phi}(t)\, \vec{a}_i(t) \ . \end{cases} \tag{A.2}$$

$$\begin{cases} \vec{s}_i(x,t) &= \cos\alpha(x,t)\, \vec{a}_i(t) + \sin\alpha(x,t)\, \vec{a}_j(t) \ . \\ \vec{s}_j(x,t) &= -\sin\alpha(x,t)\, \vec{a}_i(t) + \cos\alpha(x,t)\, \vec{a}_j(t) \ . \end{cases} \tag{A.3}$$

$$\begin{cases} \dot{\vec{s}}_i(x,t) &= (\dot{\Phi}(t) + \dot{\alpha}(x,t))\, \vec{s}_j(x,t) \ . \\ \dot{\vec{s}}_j(x,t) &= -(\dot{\Phi}(t) + \dot{\alpha}(x,t))\, \vec{s}_i(x,t) \ . \end{cases} \tag{A.4}$$

where $\dot{(\ )} \triangleq \frac{d(\ )}{dt}$. To simplify our latter development, the following assumption is made.

Assumption A.1 *The angle of flexural deflection, $\alpha(x,t)$, is uniformly small for $x \in [0, L]$, and the plane cross-section remains plane after deformation.*

With Assumption A.1, we have

$$\sin\alpha \approx \alpha \ , \qquad \cos\alpha \approx 1 \ . \tag{A.5}$$

Therefore, eq.(A.3) can be rewritten as

$$\begin{cases} \vec{s}_i(x,t) &= \vec{a}_i(t) + \alpha(x,t)\, \vec{a}_j(t) \ . \\ \vec{s}_j(x,t) &= -\alpha(x,t)\, \vec{a}_i(t) + \vec{a}_j(t) \ . \end{cases} \tag{A.6}$$

Let $\vec{p}$ be the position vector of a material particle of the beam initially located at $\vec{p}_0 = x\vec{i} + y\vec{j}$ in the reference frame. Let $u(x,t)$ and $v(x,t)$ be the movements in $\vec{a}_i$ and $\vec{a}_j$ directions, respectively, of a particle initially located at $(x,0)$ in the reference frame. We have

$$\vec{p}(x,y,t) = [x + v(x,t)]\vec{a}_i(t) + v(x,t)\vec{a}_j(t) + y\vec{s}_j(x,t) \ . \tag{A.7}$$

Also,

$$\varepsilon_x = \frac{du(x,t)}{dx} \ , \qquad \gamma(x,t) = \frac{dv(x,t)}{dx} - \alpha(x,t) \ . \tag{A.8}$$

For the notational simplicity, explicit indication of the arguments x, y, and t will often be omitted. Taking the derivative on both sides of eq.(A.7) and considering eqs.(A.2) and (A.4) result in

$$\dot{\vec{p}} = [\dot{u} - \dot{\Phi}v]\vec{a}_i + [\dot{v} + \dot{\Phi}(x+u)]\vec{a}_j - y(\dot{\Phi} + \dot{\alpha})\vec{s}_i \ . \tag{A.9}$$

The square of the norm of $\dot{\vec{p}}$ is

$$\begin{aligned} \| \dot{\vec{p}}(x,y,t) \|^2 &= \dot{\vec{p}} \cdot \dot{\vec{p}} \\ &= (\dot{u}^2 + \dot{v}^2) + y^2(\dot{\Phi} + \dot{\alpha})^2 + \dot{\Phi}^2[(x+u)^2 + v^2] + 2\dot{\Phi}[-\dot{u}v + (x+u)\dot{v}] \\ &- 2y(\dot{\Phi} + \dot{\alpha})[\dot{u} - v\dot{\Phi} + \alpha\dot{v} + \alpha\dot{\Phi}(x+u)] \ . \end{aligned} \tag{A.10}$$

A.2 Potential Energy and Kinetic Energy

In this section, we will present the expressions for system potential energy and kinetic energy. We first make some more assumption:

Assumption A.2 *The torsional effects of the beam are negligible.*

Assumption A.3 *The structural damping of the beam is negligible.*

Assumption A.4 *The load at the free end of the beam, M_t, is concentrated at the center of the cross-section, i.e., its coordinate in the reference frame is $(x,y,z) = (L,0,0)$.*

Consider Assumptions A.1–A.3, and further assume that the beam of interest has uniform cross-section with cross-sectional area A, the total potential energy of the

beam is composed of potential energy due to bending, shearing, and the axial stress, i.e. [149].

$$\Pi = \frac{1}{2}\int_0^L EI_z(\frac{d\alpha}{dx})^2 dx + \frac{1}{2}\int_0^L GA_s\gamma^2 dx + \frac{1}{2}\int_0^L EA\varepsilon_x^2 \ . \qquad (A.11)$$

where E is the Young's modulus, I_z is the second moment of area of the cross-section about the neutral axis, G is the shear modulus, γ is the shear strain, ε_x is the strain in longitudinal direction.

The total kinetic energy of the system is composed of the kinetic energy of the rotor at the pinned end of the beam, the kinetic energy of the concentrated load at the free end, and the kinetic energy of the beam excluding above two portions, i.e.,

$$K = K_r + K_{M_t} + K_b \ . \qquad (A.12)$$

where

$$K_r = \frac{1}{2}J_r\dot{\phi}^2(0,t) = \frac{1}{2}J_r[\dot{\Phi}(t) - \dot{\alpha}(0,t)]^2 \ . \qquad (A.13)$$

Since M_t is assumed concentrated at the tip of link, it has coordinate $(x,y) = (L,0)$, and its rotary effects are not considered. Therefore,

$$\begin{aligned} K_{M_t} &= \frac{1}{2}M_t \| \dot{\vec{p}}(L,0,t) \|^2 \\ &= \frac{1}{2}M_t\{[\dot{u}^2(L,t) + \dot{v}^2(L,t)] + \dot{\Phi}^2(t)[(L+u(L,t))^2 + v^2(L,t)] \\ &+ 2\dot{\Phi}(t)[-\dot{u}(L,t)v(L,t) + (L+u(L,t))\dot{v}(L,t)]\} \ . \end{aligned} \qquad (A.14)$$

The cross-sectional dimensions of the beam is shown in fig.A.3. The kinetic energy of the beam is

$$\begin{aligned} K_b &= \int\int\int_V \frac{1}{2}\rho \| \dot{\vec{p}}(x,y,t) \|^2 dV \\ &= \int_0^L \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_{-\frac{B}{2}}^{\frac{B}{2}} \frac{1}{2}\rho \| \dot{\vec{p}}(x,y,t) \|^2 dxdydz \ . \end{aligned} \qquad (A.15)$$

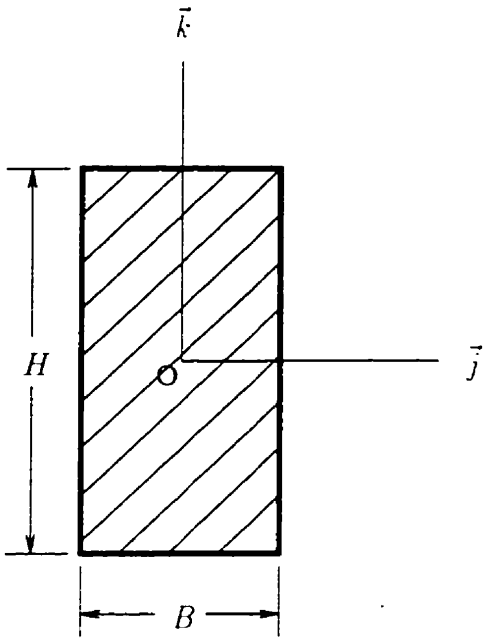


Figure A.3: Cross-Sectional Dimensions of the Beam

Use eq.(A.10) in eq.(A.15), and note $A = BH$, $I_z = \int_A y^2 dA$, we have

$$\begin{aligned} K_e &= \frac{1}{2}\int_0^L [\rho A(\dot{u}^2 + \dot{v}^2) - \rho I_z(\dot{\Phi} + \dot{\alpha})^2] dx \\ &+ \frac{1}{2}\int_0^L \rho A\{\dot{\Phi}^2[(x+u)^2 + v^2] + 2\dot{\Phi}[-\dot{u}v + (x+u)\dot{v}] \\ &\quad -2y(\dot{\Phi} + \dot{\alpha})[\dot{u} - v\dot{\Phi} + \alpha\dot{v} + \alpha\dot{\Phi}(x+u)]\} dx \,. \end{aligned} \tag{A.16}$$

The Lagrangian $\mathcal{L}$ is obtained as

$$\begin{aligned} \mathcal{L} &= K - \Pi \\ &= \frac{1}{2}J_z[\dot{\Phi}(t) - \dot{\alpha}(0,t)]^2 \\ &+ \frac{1}{2}M_z\{[\dot{u}^2(L,t) + \dot{v}^2(L,t)] + \dot{\Phi}^2(t)[(L - u(L,t))^2 + v^2(L,t)] \\ &\quad +2\dot{\Phi}(t)[-\dot{u}(L,t)v(L,t) + (L + u(L,t))\dot{v}(L,t)]\} \\ &+ \frac{1}{2}\int_0^L [\rho A(\dot{u}^2 + \dot{v}^2) + \rho I_z(\dot{\Phi} + \dot{\alpha})^2] dx \\ &+ \frac{1}{2}\int_0^L \rho A\{\dot{\Phi}^2[(x+u)^2 + v^2] + 2\dot{\Phi}[-\dot{u}v + (x+u)\dot{v}] \\ &\quad -2y(\dot{\Phi} + \dot{\alpha})[\dot{u} - v\dot{\Phi} + \alpha\dot{v} + \alpha\dot{\Phi}(x+u)]\} dx \end{aligned}$$

$$- \frac{1}{2}\int_0^L [EI_z(\frac{d\alpha}{dx})^2 + GA_s\gamma^2 + EA\varepsilon_x^2]dx \ . \tag{A.17}$$

A.3 Derivation of Equations of Motion

The equations of motion can be systematically derived by means of Hamilton's principle [107].

$$\delta\int_{t_1}^{t_2} \mathcal{L}dt + \delta\int_{t_1}^{t_2} W dt = 0 \ . \tag{A.18}$$

where W represents the work done by external forces. Here, the external force is the resultant torque T_t exerted on the rotor which will be elaborated later. Therefore,

$$W = T_t\alpha(0,t) = T_t[\Phi(t) - \alpha(0,t)] \ . \tag{A.19}$$

Using eqs.(A.17) and (A.19) in eq.(A.18) and integrating by parts, we obtain

$$\begin{aligned}
&\delta\int_{t_1}^{t_2} \mathcal{L}dt - \delta\int_{t_1}^{t_2} W dt \\
= &\int_{t_1}^{t_2}\int_0^L \{ [-\rho A\ddot{u} - \rho A\ddot{\Phi}v - 2\rho A\dot{\Phi}\dot{v} + \rho Ay(\ddot{\Phi} - \ddot{\alpha}) - \rho A\dot{\Phi}^2(x+u) \\
&\quad -\rho Ay\dot{\Phi}\alpha(\dot{\Phi} - \dot{\alpha}) - EAu'']\delta u \\
&+[-\rho A\ddot{v} - \rho A\ddot{\Phi}(x+u) - 2\rho A\dot{\Phi}\dot{u} - \rho Ay\dot{\alpha}(\dot{\Phi} - \dot{\alpha}) - \rho Ay\alpha(\ddot{\Phi} - \ddot{\alpha}) \\
&\quad +\rho A\dot{\Phi}^2 v + \rho Ay\dot{\Phi}(\dot{\Phi} - \dot{\alpha}) - GA_s\gamma']\delta v \\
&+[\rho Ay(\ddot{u} - \ddot{\Phi}v - \dot{\Phi}\dot{v} + \dot{\alpha}\dot{v} + \alpha\ddot{v} + \dot{\Phi}\dot{\alpha}(x+u) + \alpha\ddot{\Phi}(x+u) - \alpha\dot{\Phi}\dot{u} \\
&\quad -\rho I_z(\ddot{\Phi} + \ddot{\alpha}) - \rho Ay(\dot{v} + \dot{\Phi}(x+u))(\dot{\Phi} + \dot{\alpha}) + GA_s\gamma + EI_z\alpha']\delta\alpha \\
&+[-\rho I_z(\ddot{\Phi} + \ddot{\alpha}) - \rho A(\ddot{\Phi}(x+u)^2 + 2\dot{\Phi}(x+u)\dot{u} + \ddot{\Phi}v^2 + 2\dot{\Phi}v\dot{v} - \ddot{u}v \\
&\quad +(x+u)\ddot{v}) + \rho Ay(\ddot{u} - \ddot{\Phi}v - \dot{\Phi}\dot{v} + \dot{\alpha}\dot{v} + \alpha\ddot{v} + \dot{\Phi}\dot{\alpha}(x+u) \\
&\quad +\alpha\ddot{\Phi}(x+u) + \alpha\dot{\Phi}\dot{u}) - \rho Ay((\ddot{\Phi} + \ddot{\alpha})v + \dot{v}(\dot{\Phi} + \dot{\alpha}) - \dot{\alpha}(\dot{\Phi} + \dot{\alpha})(x+u) \\
&\quad -\alpha(\ddot{\Phi} + \ddot{\alpha})(x+u) - \alpha(\dot{\Phi} + \dot{\alpha})\dot{u}]\delta\Phi \ \} \ dxdt \\
+ &\int_{t_1}^{t_2} \{ [-M_t\ddot{u}(L,t) + M_t\ddot{\Phi}v(L,t) + 2M_t\dot{\Phi}\dot{v}(L,t) + M_t\dot{\Phi}^2(L + u(L,t))
\end{aligned}$$

$$
\begin{aligned}
&-EAu'(L,t)]\delta u(L,t) \\
&+[-M_t\ddot{v}(L,t) - M_t\ddot{\Phi}(L+u(L,t)) - 2M_t\dot{\Phi}\dot{u}(L,t) + M_t\dot{\Phi}^2 v(L,t) \\
&-GA_s\gamma(L,t)]\delta v(L,t) \\
&+[-J_h(\ddot{\Phi}+\ddot{\alpha}(0,t)) + EI_z\alpha'(0,t) + T_h]\delta\alpha(0,t) \\
&+[-EI_z\alpha'(L,t)]\delta\alpha(L,t) \\
&+[-J_h(\ddot{\Phi}+\ddot{\alpha}(0,t)) - M_t\ddot{\Phi}(L+u(L,t))^2 - 2M_t\dot{\Phi}(L+u(L,t))\dot{u}(L,t) \\
&-M_t\ddot{\Phi}v^2(L,t) - 2M_t\dot{\Phi}v(L,t)\dot{v}(L,t) + M_t\ddot{u}(L,t)v(L,t) \\
&-M_t(L+u(L,t))\ddot{v}(L,t) + T_h]\delta\Phi\ \}\,dt\ . \qquad \text{(A.20)}
\end{aligned}
$$

where $(\ )' \triangleq \frac{\partial(\ )}{\partial x}$, and in deriving eq.(A.20), following relationships have been used [107].

- operators δ and $\frac{\partial}{\partial t}$, as well as δ and $\frac{\partial}{\partial x}$ are commutative.
- the integrations with respect to t and x are interchangeable.
- for $t = t_1$, and $t = t_2$, $\delta u = \delta v = \delta\alpha = \delta\Phi \equiv 0$, which means that at the two instants $t = t_1$ and $t = t_2$, the true and varied paths coincide.

Since the virtual displacements δu, δv, $\delta\alpha$, and $\delta\Phi$ are independent and arbitrary, $\forall\ x \in (0, L)$ and $t \in (t_1, t_2)$, from eq.(A.17) and eq.(A.20), we have

$$
\begin{aligned}
&\rho A\ddot{u} - \rho A\ddot{\Phi}v - 2\rho A\dot{\Phi}\dot{v} - \rho Ay(\ddot{\Phi}+\ddot{\alpha}) - \rho A\dot{\Phi}^2(x+u) + \rho Ay\dot{\Phi}\alpha(\dot{\Phi}+\dot{\alpha}) \\
&\qquad -EAu'' = 0\ . \qquad \text{(A.21)} \\
&\rho A\ddot{v} + \rho A\ddot{\Phi}(x+u) + 2\rho A\dot{\Phi}\dot{u} - \rho Ay\dot{\alpha}(\dot{\Phi}+\dot{\alpha}) - \rho Ay\alpha(\ddot{\Phi}+\ddot{\alpha}) - \rho A\dot{\Phi}^2 v \\
&\qquad - \rho Ay\dot{\Phi}(\dot{\Phi}+\dot{\alpha}) - GA_s\gamma' = 0\ . \qquad \text{(A.22)} \\
&\rho Ay[\ddot{u} - \ddot{\Phi}v - \dot{\Phi}\dot{v} + \dot{\alpha}\dot{v} + \alpha\ddot{v} + \dot{\Phi}\dot{\alpha}(x+u) + \alpha\ddot{\Phi}(x-u) + \alpha\dot{\Phi}\dot{u}] - \rho I_z(\ddot{\Phi}+\ddot{\alpha}) \\
&\qquad -\rho Ay[\dot{v} + \dot{\Phi}(x+u)](\dot{\Phi}+\dot{\alpha}) - GA_s\gamma + EI_z\alpha'' = 0\ . \qquad \text{(A.23)} \\
&\int_0^L \{-\rho I_z(\ddot{\Phi}+\ddot{\alpha}) - \rho A[\ddot{\Phi}(x+u)^2 + 2\dot{\Phi}(x+u)\dot{u} + \ddot{\Phi}v^2 - 2\dot{\Phi}v\dot{v} - \ddot{u}v + (x+u)\ddot{v}]
\end{aligned}
$$

$$+\rho Ay[\ddot{u} - \ddot{\Phi}v - \dot{\Phi}\dot{v} + \dot{\alpha}\dot{v} + \alpha\ddot{v} + \dot{\Phi}\dot{\alpha}(x-u) + \alpha\ddot{\Phi}(x+u) + \alpha\dot{\Phi}\dot{u}]$$

$$-\rho Ay[(\ddot{\Phi} + \ddot{\alpha})v + \dot{v}(\dot{\Phi} + \dot{\alpha}) - \dot{\alpha}(\dot{\Phi} + \dot{\alpha})(x+u)$$

$$-\alpha(\ddot{\Phi} + \ddot{\alpha})(x+u) - \alpha(\dot{\Phi} + \dot{\alpha})\dot{u}] \} \, dx$$

$$-J_h(\ddot{\Phi} + \ddot{\alpha}(0,t)) - M_t\ddot{\Phi}(L + u(L,t))^2 - 2M_t\dot{\Phi}(L + u(L,t))\dot{u}(L,t)$$

$$-M_t\ddot{\Phi}v^2(L,t) - 2M_t\dot{\Phi}v(L,t)\dot{v}(L,t) + M_t\ddot{u}(L,t)v(L,t)$$

$$-M_t(L + u(L,t))\ddot{v}(L,t) + T_h = 0 \, . \tag{A.24}$$

Since the virtual displacements $\delta u(L,t)$, $\delta v(L,t)$, $\delta\alpha(0,t)$, and $\delta\alpha(L,t)$ are also independent and arbitrary, we have

$$M_t\ddot{u}(L,t) - M_t\ddot{\Phi}v(L,t) - 2M_t\dot{\Phi}\dot{v}(L,t) - M_t\dot{\Phi}^2[L + u(L,t)]$$

$$+EAu'(L,t) = 0 \, . \tag{A.25}$$

$$M_t\ddot{v}(L,t) + M_t\ddot{\Phi}[L + u(L,t)] + 2M_t\dot{\Phi}\dot{u}(L,t) - M_t\dot{\Phi}^2v(L,t)$$

$$-GA_s\gamma(L,t) = 0 \, . \tag{A.26}$$

$$J_h[\ddot{\Phi} - \ddot{\alpha}(0,t)] - EI_z\alpha'(0,t) - T_h = 0 \, . \tag{A.27}$$

$$EI_z\alpha'(L,t) = 0 \, . \tag{A.28}$$

Remarks:

- Equations (A.21-A.28) represent the equations of motion of the system and the natural boundary conditions. Specifically, eqs.(A.21) and (A.22) characterize the beam vibrations in longitudinal and transversal directions, respectively, eq.(A.23) corresponds to the rotary effects of the beam cross-section, and eq.(A.24) expresses the overall balance of angular momentum of the system.

- Among the natural boundary conditions, eqs.(A.25) and (A.26) govern the vibrations of the tip load in longitudinal and transversal directions, respectively, eq.(A.27) represents the angular dynamic equilibrium at the root of the beam, and eq.(A.28) corresponds to the angular dynamic equilibrium at the tip of

the beam, noting that the tip load and the motor hub are considered parts of the beam.

- The set of equations of motion are highly nonlinear and heavily coupled between flexural deformations and rigid body motion, which are too complex to be of much use in control applications, and therefore, we have to make further simplifications.

Assumption A.5 *The longitudinal deformation and its time and spatial derivatives are negligible, i.e.,*

$$u(x,t) = \dot{u}(x,t) = \ddot{u}(x,t) = u'(x,t) = u''(x,t) \approx 0 \ , \qquad \forall\ x \in [0, L] \ . \tag{A.29}$$

Since eqs.(A.21) and (A.25) characterize the longitudinal vibrations of an infinitesimal element of the beam and the tip mass, respectively. By Assumption A.5, longitudinal vibrations are neglected, so are the eqs.(A.21) and (A.25).

Assumption A.6 *The transverse deflection, $v(x,t)$, of the beam is uniformly small for $x \in [0, L]$, and its higher order terms are negligible.*

Assumption A.7 *Beam dimension in transverse direction is small, and the rotary effects of the beam cross-section are negligible.*

By Assumptions A.5–A.7, the equations of motion and the natural boundary conditions, eqs.(A.21–A.28) are simplified as,

$$\rho A\ddot{v} + \rho A x\ddot{\Phi} - \rho A v\dot{\Phi}^2 - GA_s\gamma' = 0 \ . \tag{A.30}$$

$$GA_s\gamma + EI_z\alpha'' = 0 \ . \tag{A.31}$$

$$J_h(\ddot{\Phi} + \ddot{\alpha}(0,t)) + \int_0^L \rho A[x^2\ddot{\Phi} + 2v\dot{v}\dot{\Phi} + x\ddot{v}]\, dx$$
$$+ M_t[L^2\ddot{\Phi} + 2v(L,t)\dot{v}(L,t)\dot{\Phi} + L\ddot{v}(L,t)] - T_h = 0 \ . \tag{A.32}$$

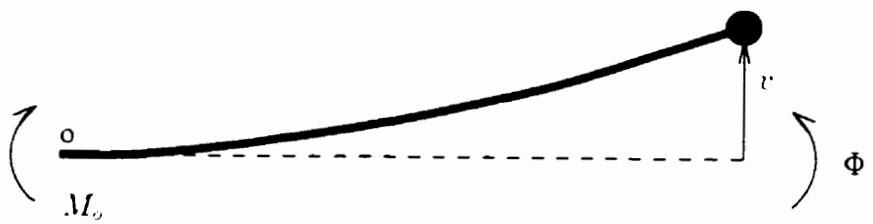


Figure A.4: Free Body Diagram of the Beam

$$M_t[\ddot{v}(L,t) + L\ddot{\Phi} - v(L,t)\dot{\Phi}^2] + GA_s\gamma(L,t) = 0 \ . \tag{A.33}$$

$$J_h[\ddot{\Phi} + \ddot{\alpha}(0,t)] - EI_z\alpha'(0,t) - T_h = 0 \ . \tag{A.34}$$

$$EI_z\alpha'(L,t) = 0 \ . \tag{A.35}$$

Remarks:

- Differentiating eq.(A.31) w.r.t. x yields

$$GA_s\gamma' = -EI_z\alpha''' \ . \tag{A.36}$$

Using eq.(A.36) in eq.(A.30) yields a new equation of motion.

$$\rho A\ddot{v} - \rho Ax\ddot{\Phi} - \rho Av\dot{\Phi}^2 - EI_z\alpha''' = 0 \ . \tag{A.37}$$

- Investigating eq.(A.32) more closely, it is found that

$$\begin{aligned} M_o &= \int_0^L \rho A[x^2\ddot{\Phi} + 2v\dot{v}\dot{\Phi} - x\ddot{v}]\,dx \\ &\quad + M_t[L^2\ddot{\Phi} + 2v(L,t)\dot{v}(L,t)\dot{\Phi} + L\ddot{v}(L,t)] \ . \end{aligned} \tag{A.38}$$

where M_o is the bending moment on the cross-section and at the root of the beam, as shown in fig.A.4 The integral part of eq.(A.38) represents the bending moment incurred by the beam without the tip load, the first and third terms of which correspond to the rigid body motion and transverse vibration, respectively, while the second term represents the Coriolis effect due to the interference between rigid body motion and transverse vibration. Similarly,

the bending moment incurred by the tip load is also composed of the same three effects.

- Using eq.(A.38) in eq.(A.32) results in

$$J_h(\ddot{\Phi} + \ddot{\alpha}(0,t)) + M_z - T_h = 0 \ . \tag{A.39}$$

We also know that [12]

$$M_z = -EI_z\alpha'(0,t) \ . \tag{A.40}$$

Consider eq.(A.40) and compare eq.(A.39) with eq.(A.34), we find that they are actually identical, and therefore, one of them, say eq.(A.39), can be omitted.

- In eq.(A.33), $GA_s\gamma(L,t)$ represents the shear force at $x = L$. From eq.(A.31), we have

$$GA_s\gamma(L,t) = -EI_z\alpha''(x,t)|_{x=L} \stackrel{\Delta}{=} -EI_z\alpha''(L,t) \ . \tag{A.41}$$

Using eq.(A.41) in eq.(A.33) results in

$$M_t[\ddot{v}(L,t) - L\ddot{\Phi} - v(L,t)\dot{\Phi}^2] - EI_z\alpha''(L,t) = 0 \ . \tag{A.42}$$

- Since the root of the beam is pinned, we have a geometric boundary condition,

$$v(0,t) = 0 \ . \tag{A.43}$$

Summarizing above discussions, we obtain the new equation of motion and boundary conditions as follows, subject to Assumptions A.1–A.7.

$$\rho A\ddot{v} + \rho Ax\ddot{\Phi} - \rho Av\dot{\Phi}^2 + EI_z\alpha''' = 0 \ . \tag{A.44}$$

$$v(0,t) = 0 \ . \tag{A.45}$$

$$M_t[\ddot{v}(L,t) + L\ddot{\Phi} - v(L,t)\dot{\Phi}^2] - EI_z\alpha''(L,t) = 0 \ . \tag{A.46}$$

$$J_h[\ddot{\Phi} + \ddot{\alpha}(0,t)] - EI_z\alpha'(0,t) - T_h = 0 \ . \tag{A.47}$$

$$EI_z\alpha'(L,t) = 0 \ . \tag{A.48}$$

where the nonlinear centrifugal terms are still present in eqs.(A.44) and (A.46).

Assumption A.8 *Shear strain of the beam, $\gamma(x,t)$, is uniformly small for $x \in [0, L]$, such that*

$$\alpha(x,t) = v'(x,t) - \gamma(x,t) \approx v'(x,t) \ , \qquad x \in [0, L] \ . \tag{A.49}$$

Assumption A.9 *The centrifugal effects due to coupling between rigid body motion and beam vibration, which are proportional to $v(x,t)\dot{\Phi}^2$, are uniformly small for $x \in [0, L]$, and are negligible.*

Using Assumptions A.8 and A.9 in eqs.(A.44–A.48), we obtain the final form of equation of motion and boundary conditions.

$$\rho A\ddot{v} - EI_z v'''' - \rho Ax\ddot{\Phi} = 0 \ . \tag{A.50}$$

$$v(0,t) = 0 \ . \tag{A.51}$$

$$v''(L,t) = 0 \ . \tag{A.52}$$

$$M_t\ddot{v}(L,t) + M_tL\ddot{\Phi} - EI_z v'''(L,t) = 0 \ . \tag{A.53}$$

$$J_h[\ddot{\Phi} + \ddot{v}'(0,t)] - EI_z v''(0,t) - T_h = 0 \ . \tag{A.54}$$

The system dynamic model, eqs.(A.50–A.54), may be simplified symbolically by defining a variable transformation.

$$w(x,t) \triangleq v(x,t) + x\Phi(t) \ . \tag{A.55}$$

where the new variable, $w(x,t)$, is the superposition of rigid body motion and flexural vibration of the beam. We have

$$\begin{cases} \ddot{w}(x,t) &= \ddot{v}(x,t) + x\ddot{\Phi}(t) \ , \\ w'(x,t) &= v'(x,t) + \Phi(t) \ , \\ w''(x,t) &= v''(x,t) \ , \\ w'''(x,t) &= v'''(x,t) \ , \\ w''''(x,t) &= v''''(x,t) \ , \\ w(0,t) &= v(0,t) \ , \\ \ddot{w}'(0,t) &= \ddot{v}'(0,t) + \ddot{\Phi}(0,t) \ . \end{cases} \tag{A.56}$$

Use eq.(A.56) in eqs.(A.50-A.54), we obtain the system dynamic model in terms of the transformed variable, $w(x,t)$.

$$\rho A\ddot{w} + EI_z w'''' = 0 \ . \tag{A.57}$$

$$w(0,t) = 0 \ . \tag{A.58}$$

$$w''(L,t) = 0 \ . \tag{A.59}$$

$$M_t\ddot{w}(L,t) - EI_z w'''(L,t) = 0 \ . \tag{A.60}$$

$$J_h\ddot{w}'(0,t) - EI_z w''(0,t) - T_h = 0 \ . \tag{A.61}$$

A.4 Remarks

- The dynamic model of the system, eqs.(A.50-A.54), is linear, and therefore, can be solved by standard mathematical approaches, such as modal expansion method. This model has been widely used in controller design for flexible link manipulators, e.g., [2, 69, 82, 96, 125, 136], just to name a few. But one has to keep in mind that this model has been obtained after severe simplifications of the physical systems, i.e., Assumptions A.1-A.9. And any *significant* deviation

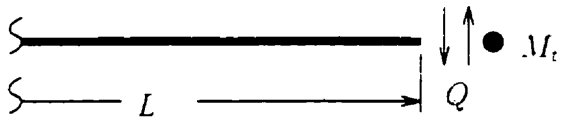


Figure A.5: Shear Force on the Free End of the Beam

of the real system from the assumed conditions could result in unsatisfactory results and cause potential problems in applications. For example, it has been shown that an uncoupled model of a beam can lead to faulty prediction of the natural frequencies of vibration [54, 202]. It has also been indicated [149] that the centrifugal effects in equations of motion have crucial role with regard to the stability of a rapidly rotating beam.

- In equation of motion of the system, eq.(A.50), the three terms correspond, respectively, to the effects of transverse vibration, shear force due to beam deflection, and rigid body motion. The first two terms correspond to the standard linear Euler-Bernoulli beam theory, while the third term accounts for the rigid body motion.

- Equations (A.52) and (A.53) indicate that, at the tip of the beam, the bending moment is zero while the shear force is not. This is the result of Assumption A.4, i.e., the tip load is concentrated at the center of the beam cross-section of the free end. Since the load is assumed to be point load, it does not incur any bending moment. But, since it has mass, it does incur a shear force on the cross-section of the beam as a result of transverse vibration and rigid body motion, as illustrated in fig.A.5, where Q represents shear force.

A.5 Eigenvalues and Eigenfunctions

To obtain the eigenvalues and eigenfunctions of the system, we let external forces vanish, i.e.,

$$T_h \equiv 0 \ . \tag{A.62}$$

Therefore,

$$\rho A \ddot{w} + EI_z w'''' = 0 \ . \tag{A.63}$$

$$w(0,t) = 0 \ . \tag{A.64}$$

$$w''(L,t) = 0 \ . \tag{A.65}$$

$$M_t \ddot{w}(L,t) - EI_z w'''(L,t) = 0 \ . \tag{A.66}$$

$$J_h \ddot{w}'(0,t) - EI_z w''(0,t) = 0 \ . \tag{A.67}$$

Assume the variable $w(x,t)$ is separable in time and space, i.e.,

$$w(x,t) = W(x)q(t) \ . \tag{A.68}$$

Use eq.(A.68) in eq.(A.63) leads to

$$\frac{EI_z}{\rho A}\frac{1}{W(x)}W''''(x) = -\frac{1}{q(t)}\ddot{q}(t) \ . \tag{A.69}$$

In eq.(A.69), the left hand side depends only on x, while the right hand side depends only on t, they must have a constant common value, say ω^2, which is also shown to be positive [139]. Denote

$$\lambda^4 \triangleq \frac{\rho A}{EI_z}\omega^2 \ . \tag{A.70}$$

We have

$$\ddot{q}(t) + \omega^2 q(t) = 0 \ . \tag{A.71}$$

$$W''''(x) - \lambda^4 W(x) = 0 \ . \tag{A.72}$$

where λ and $W(x)$ are the eigenvalue and eigenfunction of the system, respectively, and ω is the natural frequency of the system.

Use eq.(A.68) in eqs.(A.64-A.67), we have

$$W(0)q(t) = 0 \ . \tag{A.73}$$

$$W''(L)q(t) = 0 \ . \tag{A.74}$$

$$M_t W(L)\ddot{q}(t) - EI_z W'''(L)q(t) = 0 \ . \tag{A.75}$$

$$J_h W'(0)\ddot{q}(t) - EI_z W'''(0)q(t) = 0 \ . \tag{A.76}$$

Since $q(t) \not\equiv 0$, and $\ddot{q}(t) = -\omega^2 q(t)$, we have the boundary conditions

$$W(0) = 0 \ . \tag{A.77}$$

$$W''(L) = 0 \ . \tag{A.78}$$

$$M_t W(L)\omega^2 - EI_z W'''(L) = 0 \ . \tag{A.79}$$

$$J_h W'(0)\omega^2 - EI_z W''(0) = 0 \ . \tag{A.80}$$

We now solve eq.(A.72) to obtain eigenvalues and eigenfunctions. Assume

$$W(x) = ce^{sx} \ . \tag{A.81}$$

where c and s are constants. Using eq.(A.81) in eq.(A.72), we obtain the auxiliary equation

$$s^4 - \lambda^4 = 0 \ . \tag{A.82}$$

Roots of this equation are solved as

$$s_{1,2} = \pm\lambda \ , \qquad s_{3,4} = \pm i\lambda \ . \tag{A.83}$$

The solution of eq.(A.72) is obtained as

$$W(x) = c_1' e^{\lambda x} + c_2' e^{-\lambda x} + c_3' e^{i\lambda x} + c_4' e^{-i\lambda x} \ . \tag{A.84}$$

where c'_k, $k = 1.2.3.4$, are constants. Reorganize eq.(A.84), we have

$$W(x) = c_1 \cos(\lambda x) + c_2 \sin(\lambda x) + c_3 \cosh(\lambda x) + c_4 \sinh(\lambda x) \ . \qquad (A.85)$$

where c_k, $k = 1.2.3.4$, are new constants to be determined from boundary conditions. Use eq.(A.85) in eqs.(A.77-A.80), and after some manipulations we have

$$[\cos(\lambda L) + \cosh(\lambda L) - 2\alpha \sinh(\lambda L)]\ c_1 + [\sin(\lambda L) + \sinh(\lambda L)]\ c_2 = 0 \ . \qquad (A.86)$$

$$\{\sin(\lambda L) - \sinh(\lambda L) + 2\alpha \cosh(\lambda L) + \beta[\cos(\lambda L) - \cosh(\lambda L) + 2\alpha \sinh(\lambda L)]\}\ c_1$$

$$+\{-\cos(\lambda L) - \cosh(\lambda L) + \beta[\sin(\lambda L) - \sinh(\lambda L)]\}\ c_2 = 0 \ . \qquad (A.87)$$

$$c_3 = -c_1 \ . \qquad (A.88)$$

$$c_4 = 2\alpha c_1 - c_2 \ . \qquad (A.89)$$

where

$$\alpha \triangleq \frac{\beta}{(\lambda L)^3} \frac{J_t}{J_r} \ , \qquad \beta \triangleq \lambda L \frac{M_r}{M_t} \ , \qquad M_t \triangleq \rho A L \ , \qquad J_t \triangleq \frac{\rho A L^3}{3} \ . \qquad (A.90)$$

For a non-trivial solution of eq.(A.72), equations (A.86) and (A.87) have to give a non-trivial solution of the constants c_1 and c_2, and there, the determinant of their coefficients must be zero, i.e.,

$$\det \begin{bmatrix} C_{11}(\lambda) & C_{12}(\lambda) \\ C_{21}(\lambda) & C_{22}(\lambda) \end{bmatrix} = 0 \qquad (A.91)$$

where

$$\begin{cases} C_{11}(\lambda) \triangleq \cos(\lambda L) + \cosh(\lambda L) - 2\alpha \sinh(\lambda L) \ . \\ C_{12}(\lambda) \triangleq \sin(\lambda L) + \sinh(\lambda L) \ . \\ C_{21}(\lambda) \triangleq \sin(\lambda L) - \sinh(\lambda L) + 2\alpha \cosh(\lambda L) \\ \qquad\qquad +\beta[\cos(\lambda L) - \cosh(\lambda L) + 2\alpha \sinh(\lambda L)] \ . \\ C_{22}(\lambda) \triangleq -\cos(\lambda L) - \cosh(\lambda L) + \beta[\sin(\lambda L) - \sinh(\lambda L)] \ . \end{cases} \qquad (A.92)$$

eq.(A.91) gives the characteristic equation of the system.

$$\begin{aligned} & 1 + \cos(\lambda L)\cosh(\lambda L) + \alpha[\sin(\lambda L)\cosh(\lambda L) - \cos(\lambda L)\sinh(\lambda L)] \\ & \quad + \beta[\cos(\lambda L)\sinh(\lambda L) - \sin(\lambda L)\cosh(\lambda L) \\ & \quad + 2\alpha\sin(\lambda L)\sinh(\lambda L)] = 0 . \end{aligned} \tag{A.93}$$

Or, by eq.(A.90),

$$\begin{aligned} & (\lambda L)^3[1 + \cos(\lambda L)\cosh(\lambda L)] + 3\frac{J_t}{J_h}[\sin(\lambda L)\cosh(\lambda L) - \cos(\lambda L)\sinh(\lambda L)] \\ & \quad + (\lambda L)^4\frac{M_t}{M_h}[-\sin(\lambda L)\cosh(\lambda L) + \cos(\lambda L)\sinh(\lambda L)] \\ & \quad - 6(\lambda L)\frac{M_t}{M_h}\frac{J_t}{J_h}\sin(\lambda L)\sinh(\lambda L) = 0 . \end{aligned} \tag{A.94}$$

The roots of eq.(A.94), λ_i, $i = 1, 2, \ldots$, are the eigenvalues of system, of which there are infinite numbers due to the continuous nature of the system. It is easy to see that

$$\lambda_0 = 0 \tag{A.95}$$

is a solution of the characteristic equation, eq.(A.94), which corresponds to the rigid body motion of the beam, and the natural frequency $\omega_0 = 0$. In this case, the mode shape and the time dependent variable are given as

$$\begin{cases} W_0(x) = x , \\ q_0(t) = \Phi(t) . \end{cases} \tag{A.96}$$

For $\lambda_i \neq 0$, $i = 1, 2, \ldots$, i.e., for flexural modes, from eq.(A.87), we have

$$c_2 = \frac{C_{21}(\lambda_i)}{C_{22}(\lambda_i)}\, c_1 , \tag{A.97}$$

where $C_{21}(\lambda_i)$ and $C_{22}(\lambda_i)$ are given in eq.(A.92). Using eqs.(A.97), (A.88), and (A.89) in eq.(A.85), the eigenfunctions, or mode shapes, of the system are obtained

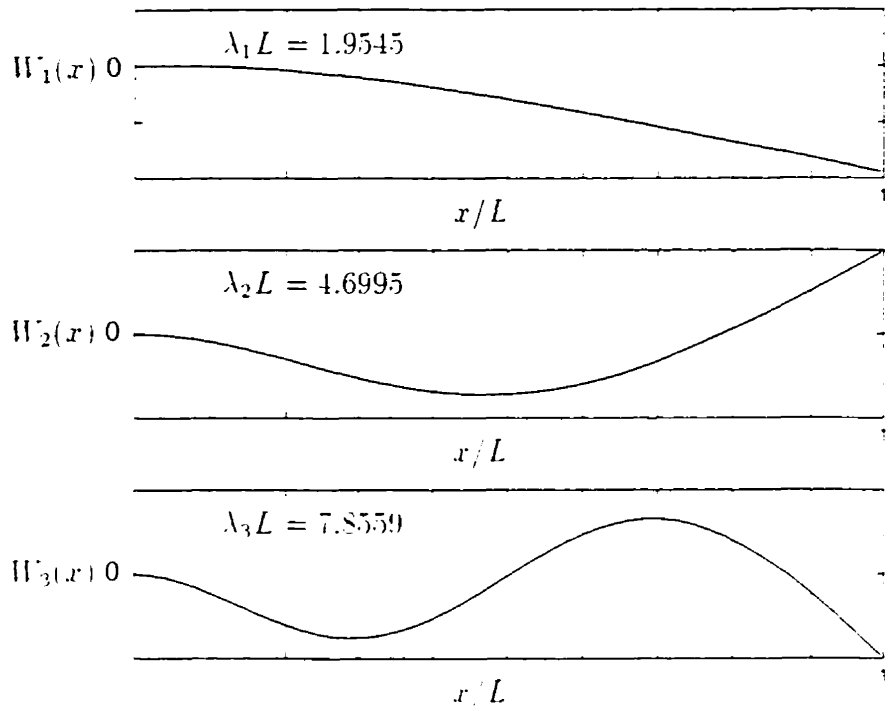


Figure A.6: First Three Flexural Eigenvalues and Eigenfunctions

as

$$W_i(x) = \{\cos(\lambda_i x) - \cosh(\lambda_i x) - 2\alpha_i \sinh(\lambda_i x) + \frac{C_{21}(\lambda_i)}{C_{22}(\lambda_i)}[\sin(\lambda_i x) - \sinh(\lambda_i x)]\} c_1 , \qquad i = 1, 2, \ldots \quad (A.98)$$

where c_1 is an arbitrary constant, and without loss of generality, can be assumed to be unity, i.e., $c_1 \hat{=} 1$. If the beam has no tip load, i.e., $M_t = 0$, the mode shape, eq.(A.98) can be shown to be the same as that of obtained in [9].

For $M_t = 0$ and $\frac{J_x}{J_h} = 0.187$, the first three flexural modes and corresponding eigenvalues, $\lambda_i L$, $i = 1, 2, 3$, are shown in fig.A.6.

A.6 Orthogonality of Eigenfunctions

It is known that for continuous systems, there are infinite number of natural frequencies, ω_i, and mode shapes, $W_i(x)$, $i = 0, 1, \ldots$, where $i = 0$ corresponds to the

rigid body motion. We now show that the mode shapes, given by eq.(A.98), and corresponding to different eigenvalues or natural frequencies, are orthogonal.

$$\int_0^L W_i(x)W_j''''(x)dx = \int_0^L W_i''(x)W_j''(x)dx + W_i(x)W_j'''(x)\Big|_0^L - W_i'(x)W_j''(x)\Big|_0^L . \tag{A.99}$$

Or

$$\int_0^L W_i''(x)W_j''(x)dx = \int_0^L W_i(x)W_j''''(x)dx - W_i(x)W_j'''(x)\Big|_0^L + W_i'(x)W_j''(x)\Big|_0^L . \tag{A.100}$$

From eqs.(A.72) and (A.77-A.80), we have

$$\begin{cases} W_j''''(x) = \lambda_j^4 W_j(x) . \\ W_j''(0) = -\lambda_j^4 \frac{J_h}{\rho A} W_j'(0) . \\ W_j''(L) = 0 . \\ W_j'''(0) = 0 . \\ W_j'''(L) = -\lambda_j^4 \frac{M_t}{\rho A} W_j(L) . \end{cases} \tag{A.101}$$

Using eq.(A.101) in eq.(A.100) yields

$$\int_0^L W_i''(x)W_j''(x)dx = \int_0^L \lambda_j^4 W_i(x)W_j(x)dx - \lambda_j^4 \frac{M_t}{\rho A} W_i(L)W_j(L) + \lambda_j^4 \frac{J_h}{\rho A} W_i'(0)W_j'(0) . \tag{A.102}$$

Similarly, we can obtain

$$\int_0^L W_j''(x)W_i''(x)dx = \int_0^L \lambda_i^4 W_i(x)W_j(x)dx + \lambda_i^4 \frac{M_t}{\rho A} W_i(L)W_j(L) + \lambda_i^4 \frac{J_h}{\rho A} W_i'(0)W_j'(0) . \tag{A.103}$$

Subtracting eq.(A.102) from eq.(A.103) results in

$$(\lambda_i^4 - \lambda_j^4)\left[\int_0^L \rho A W_i(x)W_j(x)dx + M_t W_i(L)W_j(L) + J_h W_i'(0)W_j'(0)\right] = 0. \tag{A.104}$$

Let

$$M_{ij} \triangleq \int_0^L \rho A W_i(x) W_j(x) dx + M_t W_i(L) W_j(L) + J_h W_i'(0) W_j'(0) . \quad (A.105)$$

If $\lambda_i \neq \lambda_j$, $\forall\, i \neq j$, we have

$$M_{ij} = \begin{cases} M_{ii} , & if\ i = j . \\ 0 , & if\ i \neq j . \end{cases} \quad (A.106)$$

Let

$$K_{ij} \triangleq \int_0^L E I_z W_i''(x) W_j''(x) dx . \quad (A.107)$$

Using eq.(A.102) in eq.(A.107) yields

$$\begin{aligned} K_{ij} &= \int_0^L \lambda_i^4 E I_z W_i(x) W_j(x) dx - \lambda_i^4 \frac{E I_z}{\rho A} M_t W_i(L) W_j(L) \\ &\quad + \lambda_i^4 \frac{E I_z}{\rho A} J_h W_i'(0) W_j'(0) . \end{aligned} \quad (A.108)$$

By eq.(A.70), it becomes

$$\begin{aligned} K_{ij} &= \int_0^L \omega_i^2 \rho A W_i(x) W_j(x) dx - \omega_i^2 M_t W_i(L) W_j(L) - \omega_i^2 J_h W_i'(0) W_j'(0) \\ &= \omega_i^2 M_{ij} . \end{aligned} \quad (A.109)$$

Therefore,

$$K_{ij} = \begin{cases} \omega_i^2 M_{ii} , & if\ i = j . \\ 0 , & if\ i \neq j . \end{cases} \quad (A.110)$$

A.7 Uncoupled Equations of Motion of Undamped System

Using the expansion theorem [107], we write the solution of system equations of motion, eqs.(A.57–A.61), as superposition of the mode shapes $W_i(x)$ multiplying corresponding time dependent generalized coordinates, $q_i(t)$, i.e.,

$$w(x,t) = \sum_{j=0}^{\infty} W_i(x) q_i(t) . \quad (A.111)$$

Using eq.(A.111) in eqs.(A.57), (A.60) and (A.61) results in

$$\rho A\sum_{j=0}^{\infty}\ddot{q}_j(t)W_j(x)+EI_z\sum_{j=0}^{\infty}q_j(t)W_j''''(x)=0\ . \tag{A.112}$$

$$M_t\sum_{j=0}^{\infty}\ddot{q}_j(t)W_j(L)-EI_z\sum_{j=0}^{\infty}q_j(t)W_j'''(L)=0\ . \tag{A.113}$$

$$J_h\sum_{j=0}^{\infty}\ddot{q}_j(t)W_j'(0)-EI_z\sum_{j=0}^{\infty}q_j(t)W_j''(0)=T_h\ . \tag{A.114}$$

Multiply a mode shape $W_i(x)$, $i\in\{1,2,\ldots\}$, on both sides of eq.(A.112), and integrate both sides w.r.t. x, to obtain

$$\sum_{j=0}^{\infty}\ddot{q}_j(t)\int_0^L\rho AW_j(x)W_i(x)dx+\sum_{j=0}^{\infty}q_j(t)\int_0^L EI_zW_j''''(x)W_i(x)dx=0\ . \tag{A.115}$$

Multiply $W_i(L)$ and $W_i'(0)$ on both sides of eq.(A.113) and eq.(A.114), respectively. We have

$$\sum_{j=0}^{\infty}\ddot{q}_j(t)M_tW_j(L)W_i(L)-\sum_{j=0}^{\infty}q_j(t)EI_zW_j'''(L)W_i(L)=0\ . \tag{A.116}$$

$$\sum_{j=0}^{\infty}\ddot{q}_j(t)J_hW_j'(0)W_i'(0)-\sum_{j=0}^{\infty}q_j(t)EI_zW_j''(0)W_i'(0)=W_i'(0)T_h\ . \tag{A.117}$$

Sum both sides of eqs.(A.115–A.117), respectively. We have

$$\begin{aligned}&\sum_{j=0}^{\infty}\ddot{q}_j(t)\Big[\int_0^L\rho AW_j(x)W_i(x)dx+M_tW_j(L)W_i(L)-J_hW_j'(0)W_i'(0)\Big]\\ +\ &\sum_{j=0}^{\infty}q_j(t)\Big[\int_0^L EI_zW_j''''(x)W_i(x)dx-EI_zW_j'''(L)W_i(L)-EI_zW_j''(0)W_i'(0)\Big]\\ =\ &W_i'(0)T_h\ .\end{aligned} \tag{A.118}$$

Using eqs.(A.70) and (A.101) in eqs.(A.118) results in

$$\begin{aligned}&\sum_{j=0}^{\infty}\ddot{q}_j(t)\Big[\int_0^L\rho AW_j(x)W_i(x)dx+M_tW_j(L)W_i(L)+J_hW_j'(0)W_i'(0)\Big]\\ +\ &\sum_{j=0}^{\infty}q_j(t)\Big[\int_0^L\rho A\omega_j^2W_j(x)W_i(x)dx+\omega_j^2W_j(L)W_i(L)+\omega_j^2J_hW_j'(0)W_i'(0)\Big]\\ =\ &W_i'(0)T_h\ .\end{aligned} \tag{A.119}$$

Considering eqs.(A.105) and (A.109), we have

$$\sum_{j=0}^{\infty}[M_{ij}\ddot{q}_j(t) + K_{ij}q_j(t)] = W_i'(0)T_h \ . \qquad (A.120)$$

By the orthogonality of mode shapes, eq.(A.106) and eq.(A.110), we obtain the uncoupled equations of motion

$$M_{ii}\ddot{q}_i(t) + K_{ii}q_i(t) = W_i'(0)T_h \ , \qquad i = 1, 2, \ldots . \qquad (A.121)$$

Or

$$\ddot{q}_i(t) + \omega_i^2 q_i(t) = \frac{W_i'(0)}{M_{ii}}T_h \ , \qquad i = 1, 2, \ldots . \qquad (A.122)$$

A.8 Damping Resistances

In deriving the equations of motion, eq.(A.122), we have ignored the damping resistances, of which the following types can be identified.

- Damping caused by external forces opposing the transverse vibration of the beam, such as air resistance. The resultant damping force acting on the infinitesimal element is given as [55, Chap.14]

$$f_h = c(x)\dot{w}(x,t) \ , \qquad (A.123)$$

  where $c(x)$ is a damping constant per unit length.

- Resistance to internal strain, which is expressed as [55, Chap.14]

$$\sigma_s = c_s\frac{\partial \varepsilon_x}{\partial t} \ . \qquad (A.124)$$

  where c_s is a damping constant. The resultant of the stress on a cross-section can be expressed in terms of a bending moment, M_s.

$$M_s(x) = c_s I_z \dot{w}''(x,t) \ . \qquad (A.125)$$

- Viscous damping on the rotor of the actuator at the root of the beam, which is expressed as

$$f_h = c_h \dot{w}'(0,t) \ . \tag{A.126}$$

where c_h is a damping constant.

- Damping resistance on the tip load caused by external forces opposing the transverse vibration of the tip load, e.g., air resistance, which is expressed as

$$f_{M_t} = c_{M_t} \dot{w}(L,t) \ . \tag{A.127}$$

where c_{M_t} is a damping constant.

By considering these damping resistances, and expressing the displacement $w(x,t)$ as a superposition of the mode shapes of the associated undamped system, i.e.,

$$w(x,t) = \sum_{i=0}^{\infty} W_i(x) q_i(t) \ , \tag{A.128}$$

the equations of motion can be obtained as [55, 82]

$$M_{ii}\ddot{q}_i(t) + \sum_{j=0}^{\infty} C_{ij}\dot{q}_j(t) + K_{ii}q_i(t) = W_i'(0)T_h \ , \qquad i = 1,2,\ldots \tag{A.129}$$

The second term represents the effects of the damping resistances, and this term makes the equations of motion coupled. However, for a special case,

$$\begin{cases} c(x) &= a_0 \rho A \ . \\ c_{M_t} &= a_0 M_t \ . \\ c_h &= a_0 J_h \ . \\ c_s &= a_1 E \ . \end{cases} \tag{A.130}$$

where a_0 and a_1 are constants, the damping terms can also satisfy the orthogonality relationship. The corresponding equations of motion are

$$M_{ii}\ddot{q}_i(t) + (a_0 + a_1\omega_i^2)M_{ii}\dot{q}_i(t) + K_{ii}q_i(t) = W_i'(0)T_h \ , \qquad i = 1,2,\ldots \tag{A.131}$$

Let

$$2\xi_i\omega_i = a_0 + a_1\omega_i^2 \ . \tag{A.132}$$

We have

$$M_{ii}\ddot{q}_i(t) + 2\xi_i\omega_i M_{ii}\dot{q}_i(t) + K_{ii}q_i(t) = W_i'(0)T_h \ . \qquad i = 1, 2, \ldots . \tag{A.133}$$

Or

$$\ddot{q}_i(t) + 2\xi_i\omega_i\dot{q}_i(t) + \omega_i^2 q_i(t) = \frac{W_i'(0)}{M_{ii}}\ T_h \ . \qquad i = 1, 2, \ldots . \tag{A.134}$$

where ξ_i is called the damping ratio for the ith mode.

A.9 State Space Representation

If we use the rigid body motion and the first three flexural modes to approximate the displacement, $w(x,t)$, we have

$$w(x,t) \approx \sum_{i=0}^{3} W_i(x)q_i(t) \ . \tag{A.135}$$

where $W_0(x)q_0(t)$ represents the rigid body motion, and $W_0(x) = x$, $q_0(t) = \Phi(t)$, i.e.

$$w(x,t) \approx x\Phi(t) + \sum_{i=1}^{3} W_i(x)q_i(t) \ . \tag{A.136}$$

For simplicity, we will replace the approximation symbol "$\approx$" with "$=$" throughout this work. But it should be kept in mind that the fourth and higher flexural modes are truncated in eq.(A.135).

We define system state vector and output vector as

$$\begin{cases} X \ \hat{=} \ \{\ q_0 \ \ \dot{q}_0 \ \ q_1 \ \ \dot{q}_1 \ \ q_2 \ \ \dot{q}_2 \ \ q_3 \ \ \dot{q}_3 \ \}^T \ , \\ Y \ \hat{=} \ \{\ o(0,t) \ \ \dot{o}(0,t) \ \ e(L,t) \ \ \dot{e}(L,t) \ \}^T \ . \end{cases} \tag{A.137}$$

where $o(0,t)$ represents the angular movement at the root of the beam, and $v(L,t)$ represents the flexural vibration of the tip of the beam which is the displacement of the tip of the beam less the portion of rigid body motion. We have

$$o(0,t) = w'(0,t) = \Phi(t) + \sum_{i=1}^{3} W_i'(0) q_i(t) . \tag{A.138}$$

$$v(L,t) = w(L,t) - L\Phi(t) = \sum_{i=1}^{3} W_i(L) q_i(t) . \tag{A.139}$$

The system dynamic model in state space form is

$$\begin{cases} \dot{X} = AX + BT_h , \\ Y = CX . \end{cases} \tag{A.140}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\omega_1^2 & -2\zeta_1\omega_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega_2^2 & -2\zeta_2\omega_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\omega_3^2 & -2\zeta_3\omega_3 \end{bmatrix} , \tag{A.141}$$

$$B = \begin{bmatrix} 0 & \frac{1}{M_{00}} & 0 & \frac{W_1'(0)}{M_{11}} & 0 & \frac{W_2'(0)}{M_{22}} & 0 & \frac{W_3'(0)}{M_{33}} \end{bmatrix}^T , \tag{A.142}$$

$$C = \begin{bmatrix} 1 & 0 & W_1'(0) & 0 & W_2'(0) & 0 & W_3'(0) & 0 \\ 0 & 1 & 0 & W_1'(0) & 0 & W_2'(0) & 0 & W_3'(0) \\ 0 & 0 & W_1(0) & 0 & W_2(0) & 0 & W_3(0) & 0 \\ 0 & 0 & 0 & W_1(0) & 0 & W_2(0) & 0 & W_3(0) \end{bmatrix} . \tag{A.143}$$

A.10 The Joint Friction of the Manipulator

In view of the motor, T_h is the external torque load. Let T_m be the electro-magnetic torque produced on rotor, and T_f be friction torque on rotor, so that

$$T_m = T_h + T_f \ . \tag{A.144}$$

We will model the friction T_f following the approach presented by de Wit et al [36].

$$T_f = (1 - s)T_s + sT_d \ . \tag{A.145}$$

where T_s and T_d represent *stiction* and *dynamic friction*, respectively. The variable s is used to weigh each of the two friction types, and is characterized by the following dynamic equation.

$$\tau_s \dot{s} = -s - [1 - \exp(-\frac{\dot{o}(0,t)}{\dot{o}_0(0,t)})^2] \ . \tag{A.146}$$

where τ_s is the time constant for frictional lag, $\dot{o}_0(0,t)$ is the rate of the so called Stribeck effect [36].

The dynamic friction, T_d, is given as

$$T_d = a_0 sgn[\dot{o}(0,t)] - a_2 \dot{o}(0,t) \ . \tag{A.147}$$

where a_0 is the Coulomb friction coefficient, a_2 is the viscous friction coefficient, and "*sgn*" is the Signum function defined as

$$sgn(x) \triangleq \begin{cases} 1 \ . & if \ x > 0 \ . \\ 0 \ . & if \ x = 0 \ . \\ -1 \ . & if \ x < 0 \ . \end{cases} \tag{A.148}$$

The stiction, T_s, is given as

$$T_s = \begin{cases} a_0 + a_1 \ . & if \ k_s\zeta + d_s\dot{o}(0,t) \geq a_0 + a_1 \ ; \\ k_s\zeta + d_s\dot{o}(0,t) \ . & if \ -(a_0 + a_1) < k_s\zeta + d_s\dot{o}(0,t) < a_0 + a_1 \ ; \\ -(a_0 + a_1) \ . & if \ k_s\zeta + d_s\dot{o}(0,t) \leq -(a_0 + a_1) \ . \end{cases} \tag{A.149}$$

k_s and d_s represent a stiff spring and a damper, respectively, for modeling stiction. $(\alpha_0 + \alpha_1)$ represents the break away friction torque which limits stiction. ζ is a variable representing the relative displacement from the position at which stiction occurred, and is characterized by

$$\dot{\zeta} = (1 - s)\dot{\phi}(0, t) - s\frac{1}{\tau_r}\zeta \ . \tag{A.150}$$

By considering the friction of the motor, the state space description of the system, eq.(A.140), is modified as

$$\begin{cases} \dot{X} &= AX + B(T_m - T_f) \ . \\ Y &= CX \ . \end{cases} \tag{A.151}$$

where T_m is the electro-magnetic torque of the motor, and T_f is the friction characterized by eqs.(A.145–A.150). The system model may be intuitively illustrated as in fig.A.7, where the thick lines represent vector flows. V_m represents the command voltage signal to the motor. $f_m(\cdot)$ represents the function relating the command voltage and the electro-magnetic torque, i.e.,

$$T_m = f_m(V_m, \ \phi(0, t), \ \dot{\phi}(0, t)) \ . \tag{A.152}$$

and the function f_m depends on specific motor and its operating mode. It is clear that the motor friction makes the system dynamic model nonlinear and coupled.

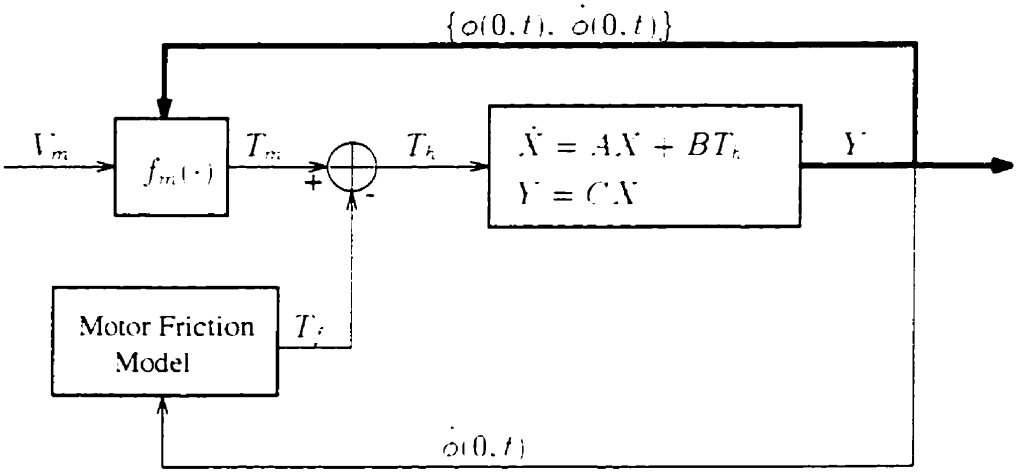


Figure A.7: Dynamic Model of a Single Link Manipulator

Appendix B

Experimental Testbed – A Single Link Robotic Manipulator

B.1 System Configuration

A flexible single link robotic manipulator has been developed in Robotics Laboratory of Directorate of Space Mechanics at Canadian Space Agency, as shown in fig.B.1. It has been serving as an experimental testbed for research on identification and control algorithms. This system is schematically shown in fig.B.2, which is composed of a personal computer, a direct drive motor with controller, an ultrasonic position and orientation sensing system, and AD/DA interface. A incremental encoder and a velometer are also installed in the motor by its manufacturer.

B.2 Direct Drive Motor and Interface to Computer

B.2.1 Introduction

The actuator of the manipulator is a Dynaserv DM1150A direct drive DC motor made by Parker Hannifin Corporation [214], which is a high torque, low-speed, outer-rotor type servo motor. A direct drive system couples the load directly to the motor without mechanical speed reducers, e.g. gears and pulleys, which can eliminate the problems associated with speed reducers, such as friction, backlash and transmission flexibility.

This motor has following parameters and features [214].

- Outer diameter : 264 mm;
- Speed : up to 2.0 rps;
- Torque : up to 370 $ft\text{-}lbs$;
- Compression loads : up to 8900 lbs;
- Overhung loads : up to 296 $ft\text{-}lbs$;
- Rotor inertia : $J_m = 0.142\ Kg \cdot m^2$
- Incremental encoder feedback with resolution of 1,024,000 $steps/rev$;
- Analog velocity loop with velocity constant of 1.2566 $\frac{rad/sec}{V}$;
- It can be operated in following control mode.

$$\text{Position Control Mode} \begin{cases} I - PD\ Type\ Position\ Control \\ P\ Type\ Position\ Control \end{cases}$$

Velocity Control Mode $\begin{cases} P\ Type\ Velocity\ Control \\ PI\ Type\ Velocity\ Control \end{cases}$

Torque Control Mode

- Accepts $\pm 10V$ for Velocity Control Mode:
- Accepts $\pm 8V$ for Torque Control Mode:

The internal control loop of the motor may be schematically shown in fig.B.3. The motor operating mode can be set by configuring the jumpers in the motor controller. The signal of the encoder is obtained by computer through an encoder reader board, which is the Model 5312 Quadrature Encoder Input - PC Card, made by Technology 80 Inc. [215]. The angular position of the motor hub in degree can be obtained as

$$\theta_m = N \cdot \frac{360}{1,024,000}\ (degree)\ . \tag{B.1}$$

where N is the number of pulses counted. The analog voltage signal of the motor velocity monitor, (VELMON), is sent into computer through a data acquisition (A/D & D/A) board, which is a RTI-815 board made by Analog Devices Inc. [216]. The angular velocity of the motor hub can be obtained as [214].

$$\dot{\theta}_m = K_{VM} \cdot (VELMON)(rad/s)\ . \tag{B.2}$$

where $K_{VM} = 1.2566 \frac{rad/s}{V}$. The command voltage of the motor, V_m, is generated in the computer and sent out to the motor controller through the same data acquisition (A/D & D/A) board. The interface between the computer and the motor controller is shown in fig.B.4. The command voltage, V_m, can be generated in two ways, with a joy-stick operated by a human operator, or with an automatic control algorithm implemented as a computer program, both of them are useful in this work.

B.2.2 Motor Operating Modes

We are only interested in Torque Control Mode and Velocity Control Mode in this work. In Torque Control Mode, the torque, T_m, is proportional to the command voltage, V_m, i.e.,

$$T_m = K_T \cdot V_m \ . \tag{B.3}$$

Consider the dynamic model for a single link manipulator derived in Appendix A, i.e., eq.A.151. For the torque control mode, it has

$$\begin{cases} \dot{X} &= AX + B(K_T V_m - T_f) \ . \\ Y &= CX \ . \end{cases} \tag{B.4}$$

where X and Y are defined in eq.(A.137), A, B, and C are defined in eqs.(A.141 - A.143), the friction T_f is defined by eqs.(A.145 - A.150). This model is illustrated in fig.B.5. In experiments, we found that the motor acceleration is too big for the flexible beam to stand. For example, $V_m = 2$ V would cause severe plastic deformation for the beam. Therefore, this mode is only used in simulation studies but not in experiments.

In P Type Velocity Control Mode, the motor angular velocity is proportional to the command voltage, i.e.,

$$\dot{o}_m = K_V V_m \ . \tag{B.5}$$

where $\dot{o}_m$ represents the angular velocity of the motor hub, K_V is a proportional constant. In this situation, the motor has an internal control loop to regulate its angular velocity to be proportional to its command voltage. The parameters of this internal loop is unknown to us, i.e., the function f_m in eq.A.152 is unknown. The dynamic model of the system in this mode is illustrated in fig.B.6. From experiments, we found that the motor velocity, $\dot{o}_m$, demonstrates complex nonlinear relationship with the motor command voltage, V_m, instead of being proportional to the command

voltage as suggested in eq.(B.5). However, in our experiments, we treat the whole system as a black box, and therefore, it is not necessary to obtain accurate system dynamic model and its parameters.

B.3 The Flexible Beam

The beam is made of stainless steel, with mass density $\rho = 7.85g/cm^3$ and Young's modulus $E = 206.7GN/m^2$ [23], and its dimensions are shown in fig.B.7. At one end, the beam is clamped to the motor hub, which can only be driven in the horizontal plane. At the other end, the beam is free and may subject to a load. With 1.28mm thickness in the transverse direction and 50mm in the vertical direction, the beam is very flexible in horizontal plane and stiff enough in the vertical plane, allowing the assumption of planar motion.

B.4 Ultrasonic Position and Orientation Sensing System

This system is the 3D Head Tracker made by Logitech Inc. [213], which includes an ultrasonic transmitter, a receiver, a control unit, power supply, and computer cable, as schematically illustrated in fig.B.8, where a picture of the ultrasound transmitter and receiver is shown in fig.B.9.

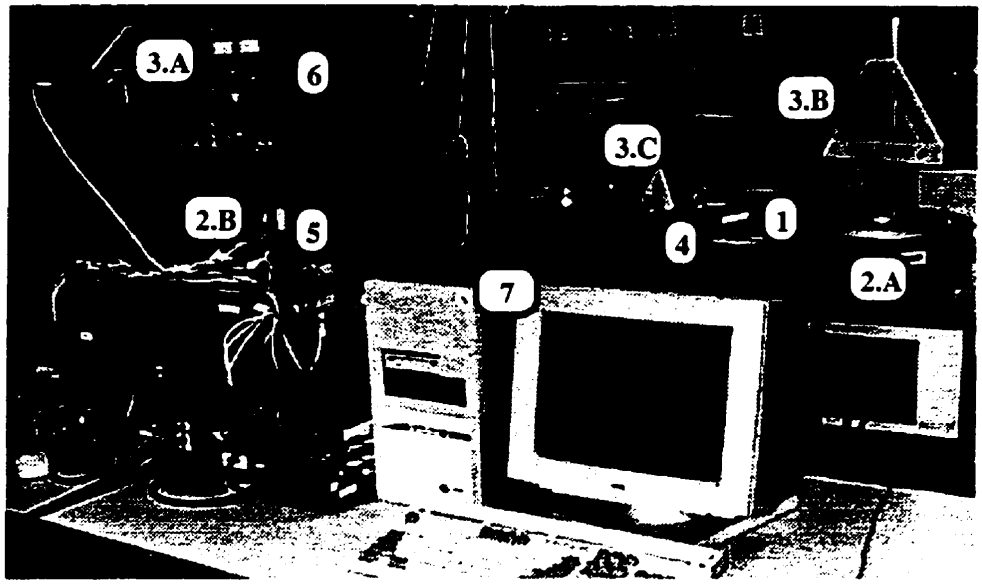
The position and orientation of the receiver can be measured and sent to computer through the serial port (RS-232). The transmitter is mounted on the hub of the motor, while the receiver is fixed at the tip of the beam, being parallel to the transmitter. The transverse deflection of the tip of the beam can be measured, with the resolution of 1/250 of an inch [213]. The measurements are updated at the speed of 50Hz, and the maximal permitted tracking speed of the receiver is 30 inches per

second [213].

The speed of transverse deflection can be obtained through numerical differentiation of the deflection, i.e.,

$$\dot{v}(t+T) \approx \frac{v(t+T)-v(t)}{T} \,. \tag{B.6}$$

where $v(t)$ is the transverse deflection of the tip of the beam at time instant t, measured with the 3D head tracker, and T is the time increment.



1. Flexible Link
2. Direct Drive Motor
   2.A. Motor
   2.B. Motor Controller
3. Ultrasonic Position and Orientation Sensing System
   3.A. Ultrasonic System Controller
   3.B. Ultrasound Transmitter
   3.C. Ultrasound Receiver
4. Tip Load of the Beam
5. Joy-stick for Manual Motor Control
6. DC Power Source
7. Personal Computer (486-100MHz) :
   Including System Controller, AD/DA Interface Board,
   and Encoder Reader Board

Figure B.1: Experimental Testbed — A Flexible Link Mechanical Manipulator

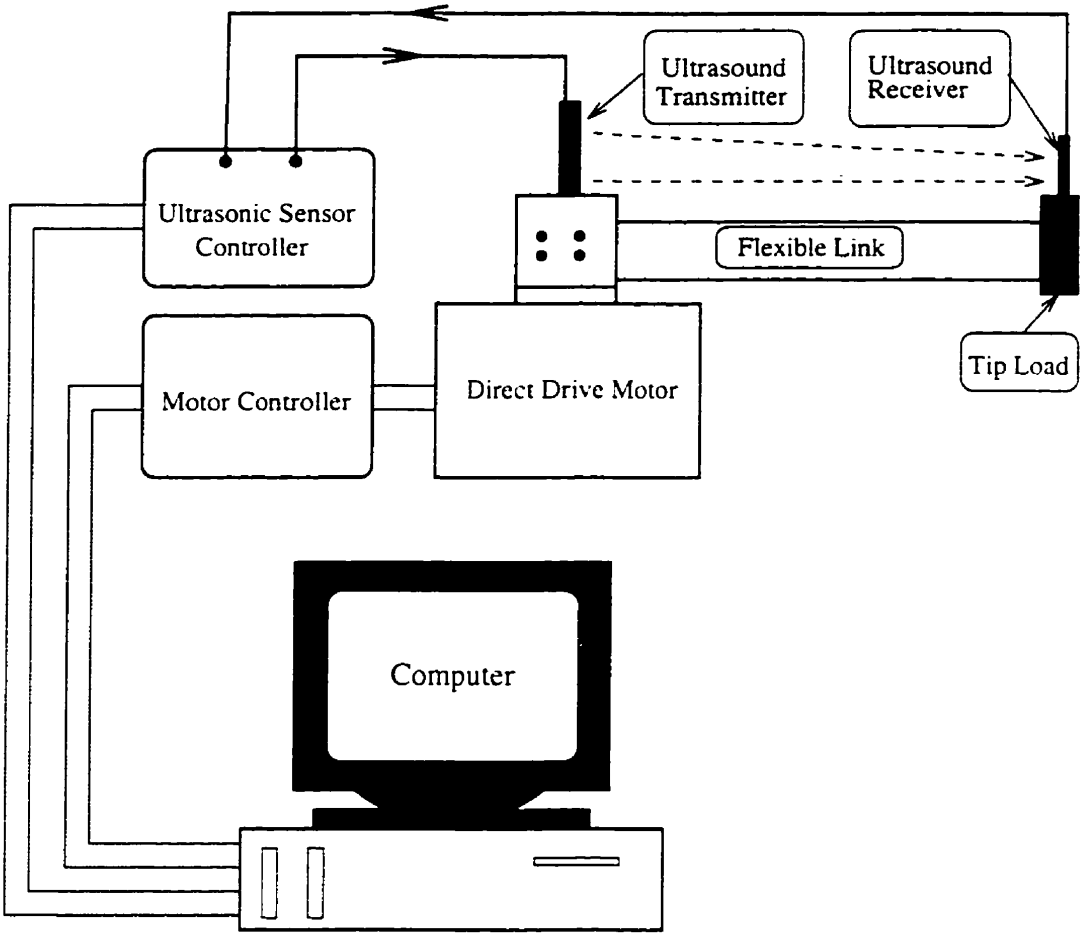


Figure B.2: Schematic Illustration of the Experimental Testbed

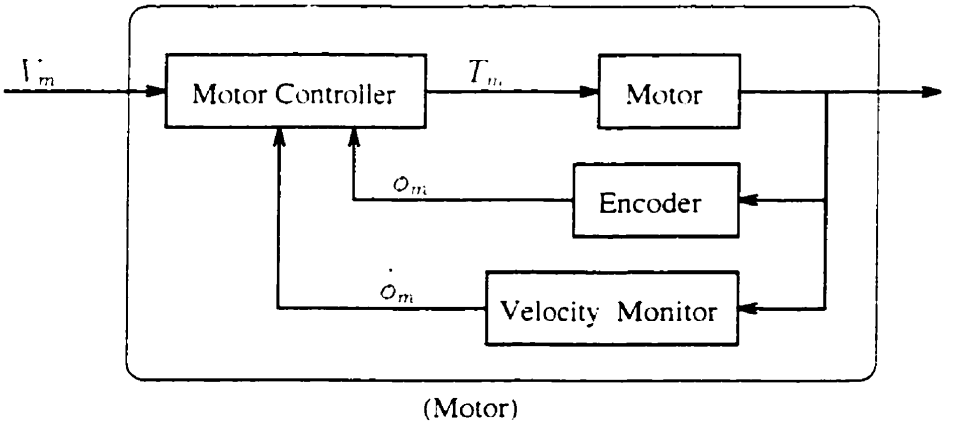


Figure B.3: Internal Control Loop of the Motor

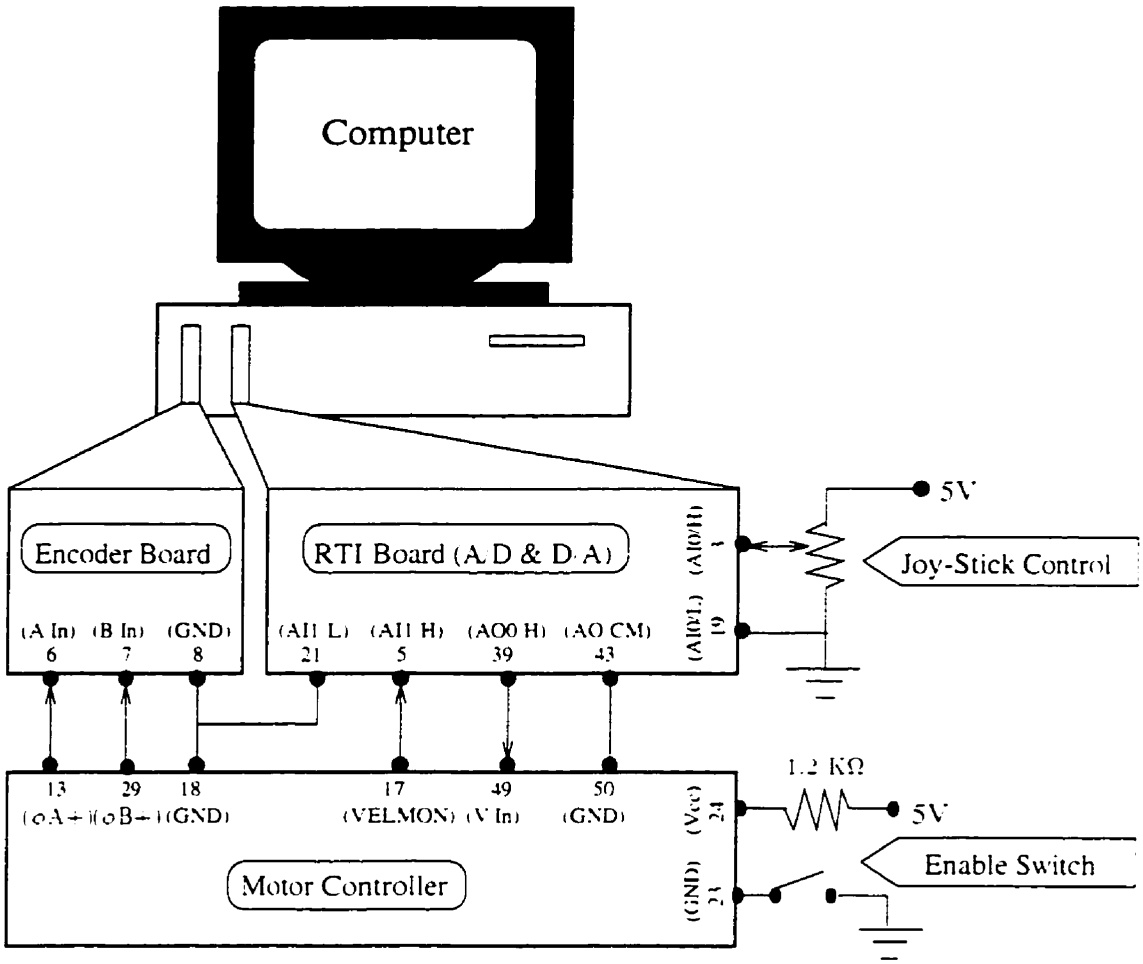


Figure B.4: Interface Between Motor Controller and Computer

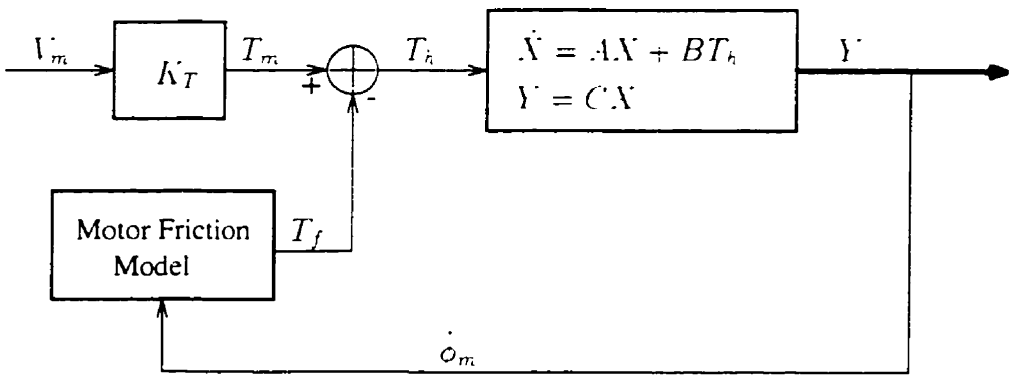


Figure B.5: Dynamic Model of the Manipulator in Torque Control Mode

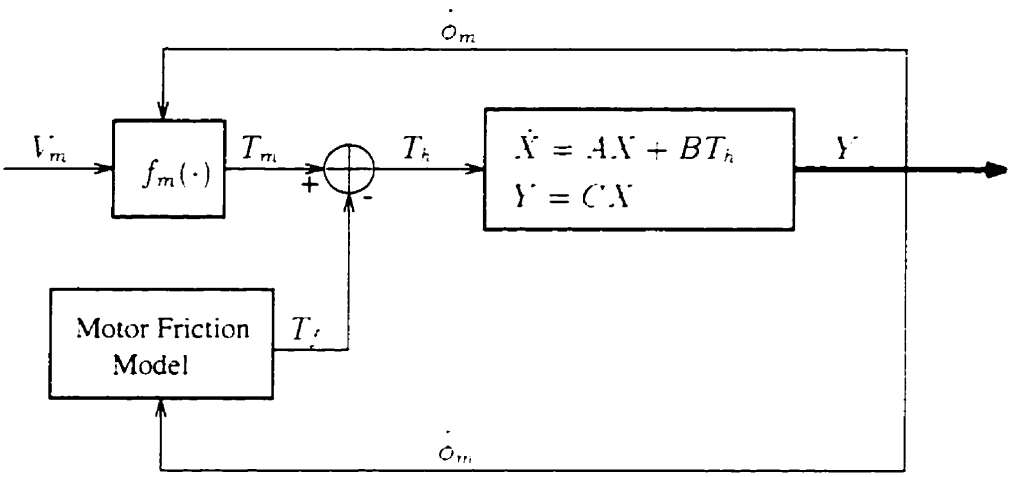


Figure B.6: Dynamic Model of the Manipulator in Velocity Control Mode

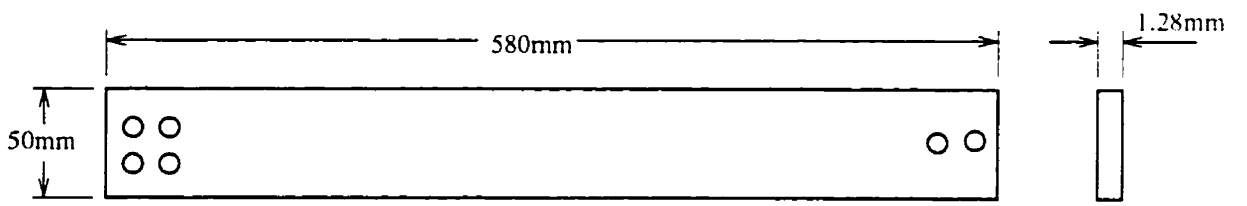


Figure B.7: Dimensions of the Beam

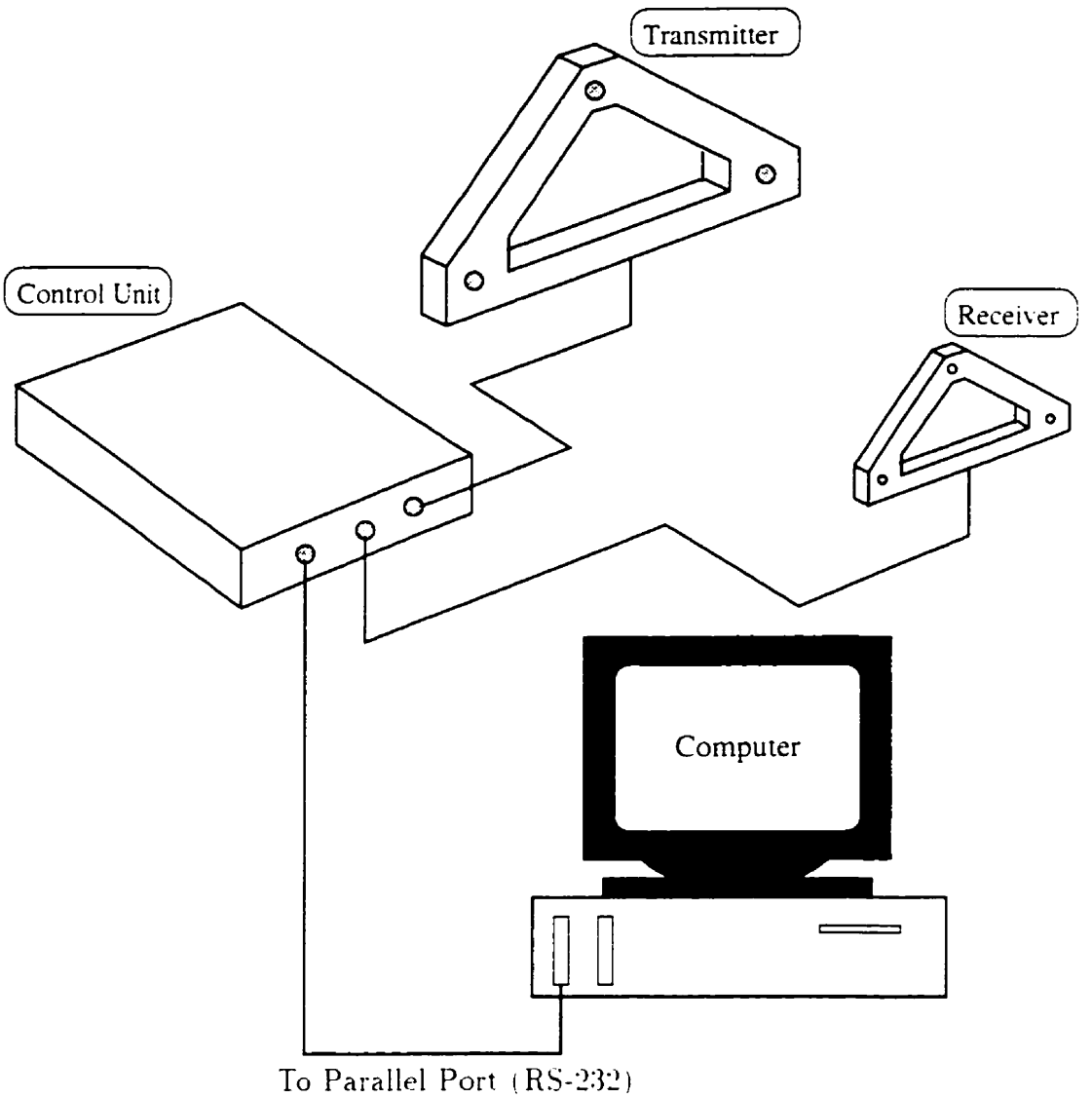


Figure B.8: Ultrasonic Position and Orientation Sensing System

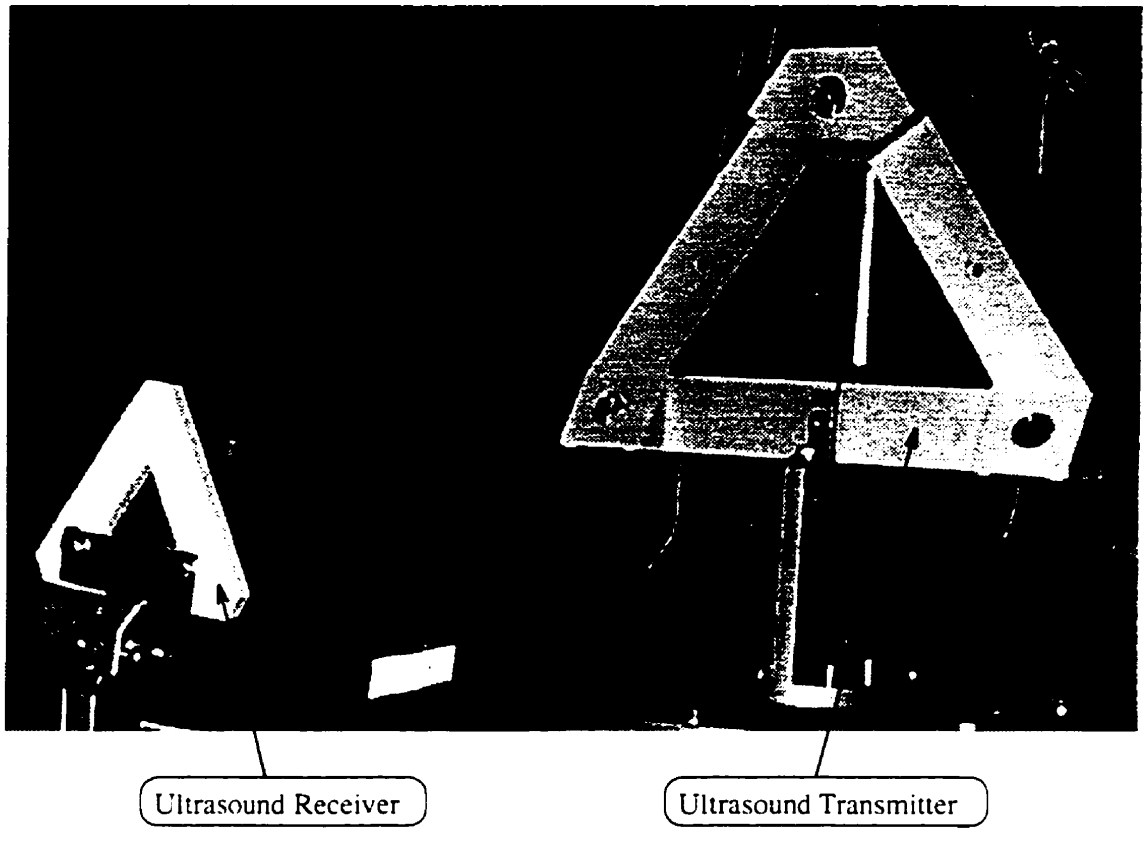


Figure B.9: Ultrasound Transmitter and Receiver

Appendix C

Thesis Subject Related Publications

Book Chapter

"Stable Identification and Adaptive Control - A Dynamic Fuzzy Logic System Approach." in *Fuzzy Evolutionary computation*. Pedrycz, W. ed. Boston, London, Dordrecht: Kluwer Academic Publishers, 1997, pp.223-248.
(with Vukovich, G.)

Journal Publication

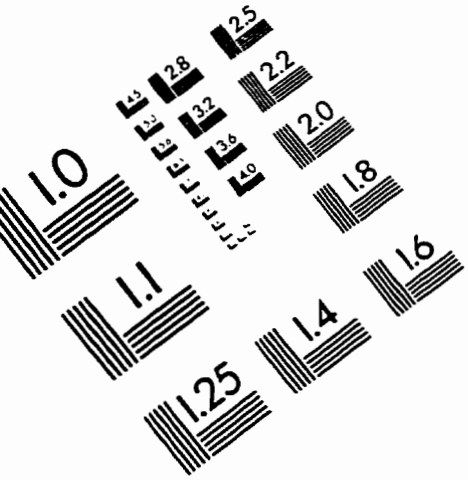
"The Dynamic Fuzzy Logic System: Nonlinear System Identification and Application to Robotic Manipulators." Journal of Robotic Systems, 14(6), 1997, pp.391-405.
(with Vukovich, G.)

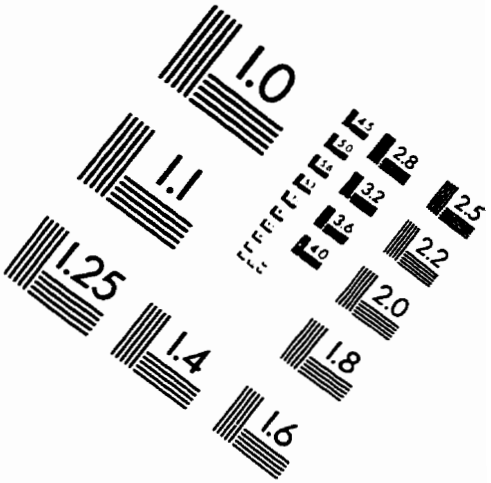
Conference Publications

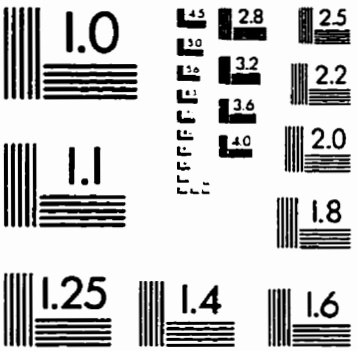
- "Identification of Nonlinear Systems - A Fuzzy Logic Approach and Experimental Demonstrations." *Proceedings of 1997 IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, Florida, Oct., 1997.
(with Vukovich, G.)

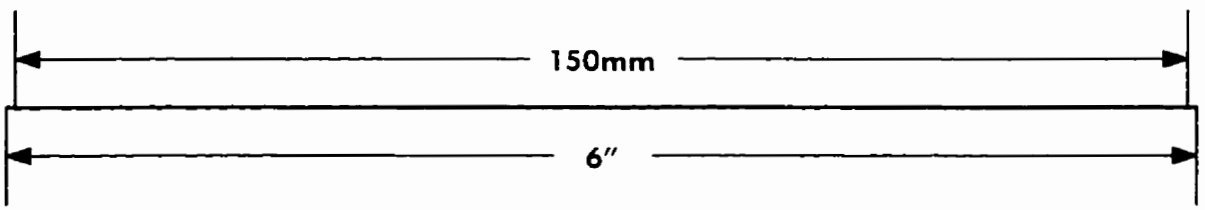
- "Fuzzy Control of A Flexible Link Manipulator." Baltimore, Maryland: *Proceedings of 1994 American Control Conference*, June 1994, pp.568-574. (with Vukovich, G., and Sasiadek, J.Z.)

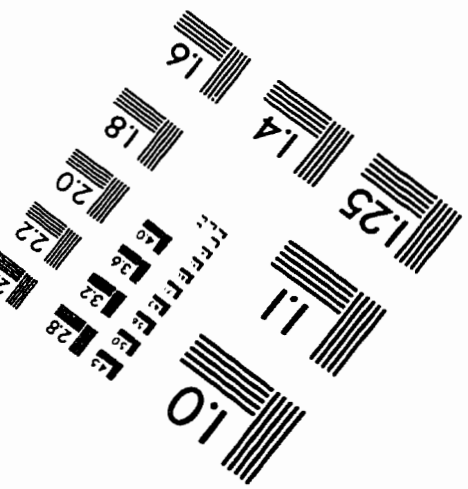
IMAGE EVALUATION TEST TARGET (QA-3)

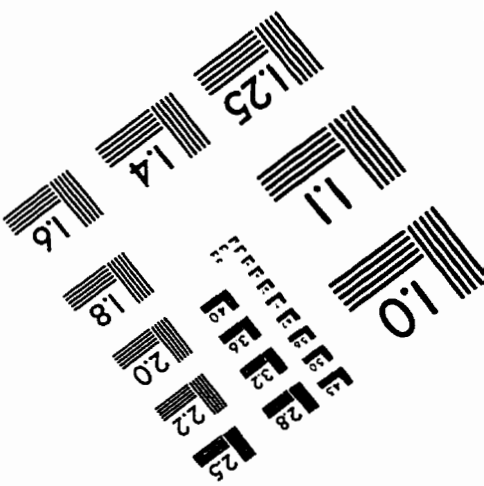












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989