



UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

H^∞ Robust Fuzzy Control for Nonlinear Dynamic Systems

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Science

By:

Hessa Jassim AL-Junaid

Supervised by:

Dr.Ebrahim Abdulla AL-Gallaf

February 2001

Examination Board



The oral examination was taken place on 26th February 2001 in the Electrical and Electronics Engineering Department meeting room (35-210) at the University of Bahrain.

Student: Hessa Jassim Al-Junaid (St. ID. 900338)
Graduate Assistant
Electrical and Electronics Engineering Department
P. O. Box 32038
University of Bahrain.
Isa Town, Bahrain.

Supervisor: Dr. Ebrahim Abdulla Al-Gallaf
Assistant Professor
Electrical and Electronics Engineering Department
P. O. Box 32038
University of Bahrain.
Isa Town, Bahrain.

Internal Examiner: Professor Mehmat Akbaba
Professor
Electrical and Electronics Engineering Department
P. O. Box 32038
University of Bahrain.
Isa Town, Bahrain.

External Examiner: Professor George Irwin
Professor
School of Electrical and Electronics Engineering
Queen's University of Belfast
Belfast BT9 5AH, UK.

I dedicate this thesis to those who are working rigorously in this fuzzy world

H^∞ Robust Fuzzy Control for Nonlinear Dynamic Systems

University of Bahrain, Department of Electrical and Electronics Engineering,

P.O. Box 32038, Isa Town, State of Bahrain.

Abstract

H^∞ robust fuzzy control systems have received a substantial amount of attention in the control research. This is due to the potential benefits of mixing a knowledge-base system representing the “fuzzy systems” with the well-established theorems in the area of “ H^∞ robust control”. This has resulted in a knowledge-based robust control system that is governed by rules in terms of describing the control aspects of a system operation.

In this sense, this thesis has presented two main issues related to H^∞ robust fuzzy control of nonlinear dynamical systems. The first has been fuzzy modeling of nonlinear dynamical systems, whereas the second was directed towards H^∞ fuzzy gain-scheduling control systems.

Regarding fuzzy modeling, that was achieved by employing Takagi-Sugeno (T-S) fuzzy modeling technique. The employed (T-S) fuzzy modeling technique was able to cluster the entire nonlinear global model into linear sub-models that are governed by fuzzy rules. (T-S) fuzzy modeling has a number of potentials with respect to other fuzzy modeling techniques; that is a small number of rules are obtained even for complicated multivariable systems.

With respect to the H^∞ fuzzy gain-scheduling, the thesis has first presented an approach for designing H^∞ fuzzy controller for disturbance rejection via defining a suitable LYAPUNOV potential function of the fuzzy model. Hence designing a controller by reducing the problem to a standard Linear Matrix Inequalities (LMI) formulation.

H^∞ fuzzy gain-scheduling has been achieved via treating the (T-S) fuzzy sub-models as a Linear Parameter Varying (LPV) system. Hence synthesizing a scheduling controller for variation in parameters.

The novel idea of H^∞ fuzzy gain-scheduling has been based on sensing variations in system parameters, adapting the LMI designed controller, while preserving the *robustness* character of the controller in phase of parametric variation.

Contents

Abstract	<i>i</i>
Acknowledgments	<i>v</i>
List of Figures	<i>vi</i>
List of Tables	<i>x</i>
Nomenclature	<i>xi</i>
 Chapter One: Introduction	
1.1 General Introduction.....	1
1.2 Artificial Neural Networks.....	3
1.3 Fuzzy Logic.....	5
1.4 Fuzzy-Neural Integrated Systems.....	6
1.5 Neurofuzzy for Modeling and Control.....	8
1.6 Classes of Intelligent Control.....	9
1.7 Robust Control.....	13
1.8 Robustness Behavior of Fuzzy Controllers.....	14
1.8.1 Literature Review in Robust Fuzzy Control.....	14
1.9 Thesis Objectives.....	18
1.10 Problem Formulation.....	18
1.11 Thesis Outline.....	18
 Chapter Two: Robust Control Essentials and Linear Matrix Inequities Formulation	
2.1 Introduction.....	21
2.2 Feedback Properties.....	22
2.3 H^∞ Control Generalization.....	25
2.4 Optimal H^∞ Control.....	29
2.5 Riccati-Based Optimal H^∞ Algorithm.....	30
2.6 H^∞ Mixed-Sensitivity Problems.....	31
2.7 Model Uncertainty.....	35
2.7.1 Uncertain State-Space Model.....	36
2.8 LMI Techniques.....	37
2.8.1 LMI Formulation of Quadratic Lyapunov Functions.....	40

2.8.2	H^∞ LMI Formulation.....	42
2.9	Conclusions.....	43

Chapter Three: Neurofuzzy Architecture and Takagi-Sugeno Fuzzy Model

3.1	Introduction.....	45
3.2	Fuzzy Sets.....	45
3.2.1	Classes of Membership Functions.....	46
3.3	Rule-Based Fuzzy Models.....	49
3.3.1	Linguistic Fuzzy Model.....	49
3.3.2	Takagi-Sugeno Model.....	56
3.4	Artificial Neural Networks.....	59
3.4.1	Multilayer Preceptron.....	59
3.4.2	Backpropagation Learning Algorithm.....	62
3.5	ANFIS: Adaptive Neurofuzzy Inference Systems.....	68
3.5.1	ANFIS Architecture.....	70
3.5.1	ANFIS Learning Algorithm.....	71
3.6	Conclusions.....	72

Chapter Four: Takagi-Sugeno Fuzzy Clustering and Modeling of Nonlinear Systems

4.1	Introduction.....	73
4.2	Overview of Modeling Terminology.....	74
4.3	Practical Relevance of Fuzzy Modeling.....	76
4.4	Modeling of Dynamic Systems.....	77
4.5	Building Fuzzy Models.....	79
4.5.1	Structure and Parameters.....	79
4.5.2	Knowledge-Based Design.....	80
4.5.3	Data-Driven Acquisition/Tuning of Fuzzy Models.....	80
4.6	NeuroFuzzy Modeling.....	82
4.7	Fuzzy Clustering.....	84
4.8	Extraction of a Linear State Space Model.....	90
4.9	Fuzzy Model Validation.....	92
4.9.1	Validation via Hypothesis Testing.....	92
4.9.2	Model Selection Criteria.....	94
4.9.3	Correlation Tests.....	97
4.10	Case studies: Neurofuzzy Modeling of Nonlinear System.....	99

4.11	Conclusions.....	116
------	------------------	-----

Chapter Five: Fuzzy H^∞ Robust Controller Synthesis for Nonlinear Systems

5.1	Introduction.....	117
5.2	Robust Fuzzy Control.....	117
5.2.1	Takagi-Sugeno Fuzzy Model Formulation.....	118
5.2.2	Basic Stability Condition.....	120
5.2.3	Fuzzy Controller.....	121
5.2.4	Closed Loop Dynamics of Fuzzy Systems.....	122
5.2.5	H^∞ Control.....	124
5.2.6	H^∞ Control Design via Fuzzy Linear Control.....	125
5.2.7	H^∞ LMI Formulation.....	128
5.3	Case Studies on the H^∞ Robust Control Synthesis.....	130
5.4	Conclusions.....	143

Chapter Six: Fuzzy-Based H^∞ Robust Gain-Scheduled Controller

6.1	Introduction.....	144
6.2	Takagi-Sugeno Fuzzy Model.....	144
6.3	Linear Parameter-Varying Systems (LPV).....	147
6.4	Robust H^∞ Control of Linear Parametric Varying (LPV) Systems	148
6.5	LMI Formulation of LPV H^∞ Controller.....	152
6.6	Fuzzy Controller.....	154
6.7	Fuzzy H^∞ Gain-Scheduler.....	156
6.8	Case Studies.....	162
6.9	Conclusions.....	178

Chapter seven: Concluding Remarks 179

Appendices

A-1	Mass-Spring-Damper System.....	182
A-2	Antenna System.....	184

References 188

List of Figures

1.1	Correspondence between a biological neuron and an artificial neuron.....	4
1.2	A direct learning controller.....	10
1.3	A learning predictive control architecture.....	10
1.4	A model reference control architecture.....	12
1.5	An internal model control architecture.....	12
2.1	Feedback Configuration.....	23
2.2	Representation of Linear Fractional Transformation.....	26
2.3	Desired loop gain.....	32
2.4	Standard feedback configuration with weights.....	34
2.5	Parameter Box.....	38
2.6	Polytopic parameter range.....	38
3.1	Temperature Membership functions.....	47
3.2	Four classes of parameterized MFs.....	47
3.3	Basic architecture of a fuzzy logic system.....	51
3.4	A schematic representation of Mamadani FIS.....	51
3.5	Different partitions of the antecedent space.....	55
3.6	Takagi–Sugeno fuzzy model as a smoothed piece-wise linear approximation of nonlinear function.....	58
3.7	A T-S fuzzy model with affine consequents.....	58
3.8	An Artificial Neural Network with two hidden layers.....	61
3.9	Possible activation functions.....	61
3.10	Three-layer back-propagation neural network.....	63
3.11	A two-input first-order Sugeno fuzzy model with two rules and an equivalent ANFIS architecture.....	69
4.1	A generic fuzzy system with fuzzification and defuzzification.....	78
4.2	Neurofuzzy network architecture with two rules	83
4.3	Flow-chart of the neurofuzzy learning procedure.....	85
4.4	Identification by fuzzy clusters.....	87
4.5	Input-output training pattern of mass-spring-damper system.....	102
4.6	Input-output training pattern of mass-spring system prior to clustering.....	102

4.7	Input-output training pattern of mass-spring system after clustering.....	103
4.8	Extracted membership functions of all regressors of mass-spring fuzzy model.....	103
4.9	Fuzzy model response in comparison with mass-spring system actual output...	104
4.10	Error between fuzzy model response and actual output of mass-spring system.....	104
4.11	Histogram of error distribution of mass-spring system.....	105
4.12	Modeling error auto-correlation of mass-spring system.....	105
4.13	Cross-correlation of mass-spring system input and error.....	106
4.14	Input-output training pattern used in the identification of the antenna system..	109
4.15	Input-output training pattern of antenna system prior to clustering.....	109
4.16	Input-output training pattern of antenna system after clustering.....	110
4.17	Extracted membership functions of all regressors of antenna fuzzy model associated with azimuth output.....	110
4.18	Extracted membership functions of all regressors of antenna fuzzy model associated with elevation output.....	111
4.19	Fuzzy model responses (<i>azimuth</i> and <i>elevation</i>) compared to the antenna system actual outputs.....	111
4.20	Error between fuzzy model outputs and antenna system actual outputs.....	112
4.21	Histogram of error distribution of antenna system.....	112
4.22	Cross-correlation of first input signal of antenna system against its associated error.....	113
5.1	Bode Plot of open loop fuzzy sub-model 1 of mass-spring system	133
5.2	Maximum singular value of open loop fuzzy sub-model 1 of mass-spring system	133
5.3	Impulse responses of the fuzzy model of mass-spring system due to different disturbance levels.....	134
5.4	Impulse responses of the fuzzy model of mass-spring system due to different disturbance levels, when $D_{12}=1$	134
5.5	Impulse responses of the fuzzy sub-model 1&2 of mass-spring system due to different disturbance levels.....	135
5.6	Frequency response of the closed loop fuzzy sub-model 1 of mass-spring system.....	136

5.7	Maximum singular value of closed loop fuzzy sub-model 1 of mass-spring system	136
5.8	Bode Plot of open loop fuzzy sub-model 1 of antenna system for its both inputs	140
5.9	Maximum singular values of open loop fuzzy sub-model 1 of antenna system.	140
5.10	Impulse responses of closed loop fuzzy model of antenna system	141
5.11	Frequency responses of the controlled fuzzy sub-model 1 of antenna system.	142
5.12	Maximum singular values of closed loop fuzzy sub-model 1 of antenna system	142
6.1	LTI Control of a LPV system	149
6.2	LPV Control of a LPV system	149
6.3	LPV Control of a mixed system	149
6.4	H^∞ Synthesis structure for LPV systems	153
6.5	A dynamic Output Feedback H^∞ Fuzzy Controller	157
6.6	Fuzzy gain-scheduling mechanism.....	161
6.7	State-space parametric variation over the entire operation region for the mass-spring system	166
6.8	Evolution of system poles as parameters vary	166
6.9	Step response of mass-spring system subject to local simulation of 50 points	167
6.10	Predefined parameter trajectory	167
6.11	Step response of the mass position over a defined trajectory parameter	168
6.12	Control action for the mass-spring system	168
6.13	Parameter variation of the linearized antenna system parameters.....	173
6.14	Parameter variation of the linearized antenna system parameters.....	174
6.15	Evolution of system poles as the parameters	175
6.16	Step response of the antenna system subject to local simulation of 50 points ..	175
6.17	Predefined parameter trajectory for the antenna system	176
6.18	Step response of the antenna system	176
6.19	Control action for the antenna system	177
A1.1	Mass-spring-damper system.....	182
A1.2	A block diagram of LFT representation of the mass-spring system.....	183
A2.1	Schematic of an antenna system.....	185
A2.2	A detailed schematic of an antenna system.....	185
A2.3	A block diagram of LFT representation of the antenna system.....	187

List of Tables

1.1	A historical sketch of soft computing constituents and conventional AI approaches.....	2
1.2	Comparison of Fuzzy Systems (FS) with Artificial Neural Networks (ANN).....	7
4.1	Different modeling paradigms.....	75
A2-1	A nonlinear Antenna parameters.....	184

Nomenclature

Chapter Two:

γ	Disturbance rejection level.
$G(s)$	A dynamic plant transfer function.
r	Reference commands.
d_i	Disturbance at plant input.
d	Output disturbances.
n	Measurement noise .
u_p	Plant input.
u	Control signal.
y_c	System output for control.
$K(s)$	Feedback controller gain.
L_i, L_o	Input and output loop transfer matrix.
S_i, S_o	Input and output sensitivity matrix.
T_i, T_o	Input and output complementary sensitivity matrices.
σ	Singular value.
$P(s)$	Plant represented in Linear Fractional Transformation LFT form.
$\mathbf{w}$	Exogenous signals in LFT.
$\mathbf{y}$	Measured output in LFT.
$\mathbf{z}$	Controlled signals in LFT.
$F_l(\mathbf{P}, \mathbf{K})$	Closed loop transfer function from $\mathbf{w}$ to $\mathbf{z}$ given by LFT.
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}$	State space matrices.
$\mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_{11}, \mathbf{D}_{12}, \mathbf{D}_{21}, \mathbf{D}_{22}$	State-space matrices of LFT representation.
X^∞, Y^∞	Solutions to the Algebraic Riccati Equation.
$\mathbf{P}$	Positive-definite quadratic LYAPUNOV Function.
$\sup$	Least upper bound.
$\mathbf{H}$	Hamiltonian matrix.
ρ	Tradeoff between good disturbance rejection at the output and control effort.
λ	Eigenvalue of a matrix.
$W_r, W_{us}, W_{ds}, W_{cs}, W_n, W_i$	Weighting functions used in feedback configuration.
$\mathbf{S}_{\text{sys}}$	System matrix.
$\delta(t)$	Vector of varying parameters.
$\text{Ric}(\mathbf{H})$	Stabilizing solution of an ARE.
$\mathbf{S}$	Convex combination of system matrices.
α	Polytopic coordinates of $\mathbf{S}$.

Chapter Three:

$\mu_A(x)$	Grade of membership function, where A represents fuzzy label.
N_i	The number of linguistic terms of the i^{th} antecedent variable.
U, V, W	Universe of discourse.
F_{inp}	Fuzzy input sets
F_{out}	Fuzzy output sets
y_c	Model output by center of gravity.
r	Number of rules.
$a(x), b(x)$	Convex linear combinations of the consequent parameters a_i and b_i
N_{layer}	Number of layers in an ANN.
$(\mathbf{x}^{(k)}, \mathbf{d}^{(k)})$	Training set of input-output pairs.
$a(\cdot)$	Activation function of ANN.
$\mathbf{y}^{(k)}$	Actual output of a network.
m	Number of Processing elements in an input layer.
n	Number of Processing elements in an output layer.
l	Number of Processing elements in a hidden layer.
$E(w)$	Cost function.
δ_{oi}	Error signal, O_i is the i^{th} node in the output layer.
net_i	Net input for PE i of the output layer.

Chapter Four:

N	Number of input-output data pairs.
(x_i, y_i)	Input-output data pairs.
a_i, b_i	Consequent parameters.
Γ_i	Diagonal matrix with normalized membership degree.
θ	Parameter vector.
J	Objective function.
c	Number of fuzzy clusters.
v_i	Fuzzy cluster centers.
$\in$	Termination criteria.
$\mathbf{u}(t)$	Vector of system's input at time t .
$\mathbf{y}(t)$	Vector of system's outputs at time t .
$\mathbf{x}(t)$	Vector of system states.

Chapter Five:

$\mathbf{w}(t)$	Vector of unknown disturbances.
$\mathbf{z}(t)$	Vector of fuzzy label represent parameter components.
$\mu_i(\mathbf{z}(t))$	Truth value for an i^{th} rule.
M_{ig}	Fuzzy membership function associated with i^{th} rule and g^{th} parameter component.
$M_{ij}(z_j(t))$	Grade of membership of $z_j(t)$.
$h(z(t))$	Firing probability for an i^{th} rule.
$\Delta f, \Delta g$	Approximation error between actual system and fuzzy model.
θ	Positive definite weighting matrix.
t_f	Terminal time of control.

Chapter Six:

$\delta(t)$	Vector of scheduling variables (varying parameter).
$\mathbf{A}(\delta(t)), \mathbf{B}(\delta(t)), \mathbf{C}(\delta(t)), \mathbf{D}(\delta(t))$	State-space of the LPV system.
$\mathbf{A}_K(\delta(t)), \mathbf{B}_K(\delta(t)), \mathbf{C}_K(\delta(t)), \mathbf{D}_K(\delta(t))$	State-space of the scheduled controller.
Θ	A polytope of matrices.
L	Number of vertices in a polytope.
$\mathbf{N}_R, \mathbf{N}_S$	Bases of null space of $(\mathbf{B}_2^T, \mathbf{D}_{12}^T)$ and $(\mathbf{C}_2, \mathbf{D}_2)$, respectively.
$\mathbf{V}$	Feedforward vector gain.
$\mathbf{A}_w(\delta(t)), \mathbf{B}_w(\delta(t)), \mathbf{C}_w(\delta(t)), \mathbf{D}_w(\delta(t))$	State-space of the entire fuzzy system.

Acronyms

AI	Artificial Intelligent.
ANN	Artificial Neural Network.
FL	Fuzzy Logic.
ANFIS	Adaptive NeuroFuzzy Inference System.
CANFIS	Coactive NeuroFuzzy Inference System.
FALCON	Fuzzy Adaptive Learning CONTROL.
FS	Fuzzy System.
FLC	Fuzzy Logic Controller.
T-S	Takagi-Sugeno fuzzy model.
LMI	Linear Matrix Inequalities.
ARE	Algebraic Riccati Equation.
LFT	Linear Fractional Transformation.
LQG	Linear Quadratic Guassian.
FLC	Fuzzy Logic Controller.
FIS	Fuzzy Inference System.
MISO	Multi-Input Single-Output system.
MIMO	Multi-Input Multi-Output system.
RBFS	Radial Basis Function NetworkS.
LPV	Linear Parameter Varying System.
LTI	Linear Time Invariant system.
LTV	Linear Time Varying System.
MF	Membership Function.
PE	Processing Elements.
FCM	Fuzzy C-Mean Clustering.
p.d.f	probability distribution function.
PDC	Parallel Distributed Compensation.
RMS	Random Mean Squares.

Chapter One:

Thesis Introduction

1.1 General Introduction

Increasingly, control systems are required to have high dynamic performance and robust behaviors, yet are expected to cope with more complex, uncertain and highly nonlinear dynamic systems. Along with this increased systems complexity is the increased abstraction and uncertainty in the models and their mathematical representation. One significant approach in dealing with major changes and uncertainty in nonlinear dynamical systems is through *intelligent modeling and control*. Intelligent controllers are generally self-organizing or adaptive and are naturally able to cope with significant changes in plants and their environment, while satisfying the control design requirements. As with any advanced control theory, a central issue is the representation and development of appropriate systems models with known approximation errors. As processes increase in complexity, they become less amenable to direct mathematical modeling that are based on physical laws, since they may be [1]:

- Distributed, stochastic, nonlinear and time varying,
- Subject to large unpredictable environmental disturbances,
- Have variables that are difficult to measure, have unknown causal relationships, or are too difficult or expensive to evaluate in real-time.

One aspect of *intelligent control* is to develop systems that incorporate the creative, abstract and adaptive attributes of humans; while minimizing the undesirable aspects such as unpredictability, inconsistency, fatigue, subjectivity and temporal instability. Intelligent controllers should be able to perform under significant systems uncertainties and incompleteness in the system and its environment. This is achieved while being both robust and reconfigurable to cope automatically with system failures and being adaptive to cope with new goals.

Research in intelligent systems started in the forties of this century [2], as presented in Table [1.1]. Hence, within the field of *Artificial Intelligent* (AI) for control applications, two important paradigms are worth mentioning. These are *Neural Networks* and *Fuzzy Logic*. Neural networks represent models of the human brain, while fuzzy logic deals with vagueness, imprecision, and linguistic approach to human reasoning. The integration between these two paradigms is the *Neurofuzzy systems* that aim at providing fuzzy systems with automatic tuning abilities.

Decade	Conventional AI	Neural Networks	Fuzzy Systems	Other Methodologies
1940s	1947 Cybernetics	1943 McCulloch-Pitts neuron model		
1950s	1956 Artificial Intelligence	1957 Perceptron		
1960s	1960 Lisp language	1960s Adaline Madaline	1965 Fuzzy sets	
1970s	mid-1970s Knowledge Engineering (expert system)	1974 Birth of Back-propagation algorithm 1975 Cognitron Neocognitron	1974 Fuzzy controller	1970s Genetic algorithm
1980s		1980 Self-organizing map 1982 Hopfield net 1983 Boltzmann machine 1986 Backpropagation algorithm boom	1985 Fuzzy modeling (T-S model)	mid-1980s Artificial Immune modeling
1990s			1990s Neuro-fuzzy modeling 1991 ANFIS 1991 FALCON 1994 CANFIS	1990 Genetic programming

Table [1.1] A historical sketch of soft computing constituents and conventional AI paradigms.

1.2 Artificial Neural Networks (ANNs)

BIOLOGICAL NEURONS

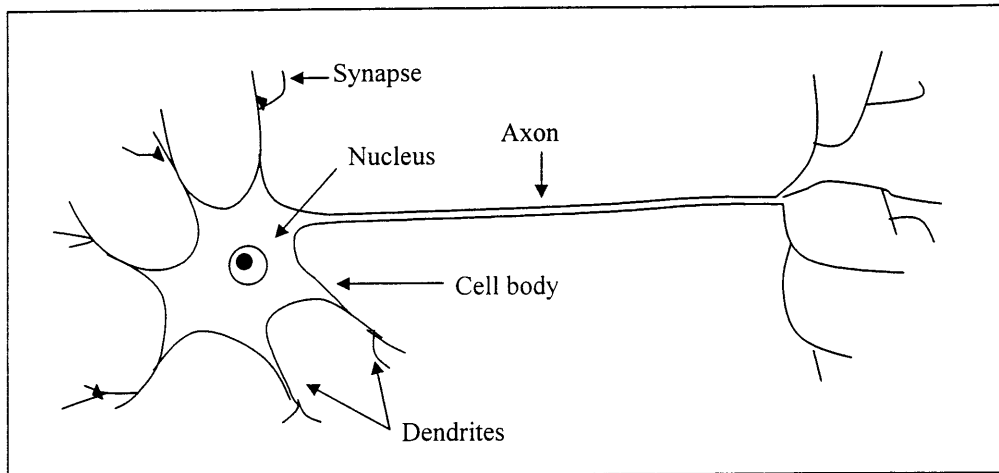
Artificial Neural Networks form a class of systems that is inspired by biological neural networks in the brain. A human brain consists of approximately 10^{11} neurons of many different types. A schematic diagram of a typical biological neuron is shown in Figure [1.1a]. A typical neuron has three major parts: the cell body or *soma*, where the cell nucleus is located, the *dendrites*, and the *axon*. Dendrites are treelike networks of nerve fiber connected to the cell body. An axon is a single, long, clindric connection extending from the cell body and carrying impulses from the neuron. The end of an axon splits into strands or a fine arborization. Each strand terminates in a small bulb-like organ called a *synapse*, where the neuron introduces its signal to the neighboring neurons. The receiving ends of these junctions on the neighboring neurons can be found both on the dendrites and on the cell bodies themselves. There are approximately 10^4 synapses per neuron in the human brain.

ARTIFICIAL NEURONS

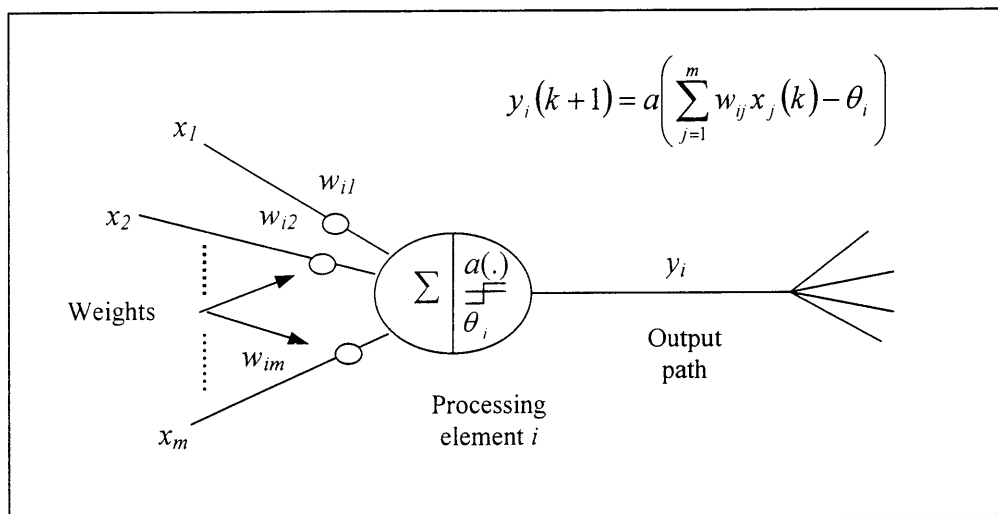
Figure [1.1b] shows a simple mathematical model of the biological neuron proposed by McCulloch and Pitts in (1943). In this model, the i^{th} processing element computes a weighted sum of its inputs x_i , while outputs $y_i=1$ (firing) or 0 (not firing) according to whether this weighted input sum is above or below a certain value of threshold θ_i .

The collective behavior of ANN, like a human brain, demonstrates the ability to *learn*, *recall*, and *generalize* from training patterns or data. Models of neural networks are specified by three basic entities: models of the processing elements themselves, models of interconnections and structures (network topology), and the learning rules (ways information is stored in the network).

The ANNs consist of a large number of simple processing elements called nodes (or neuron), that are interconnected to each other. In most cases one or more layers of neurons are considered that are connected in feedforward or recurrent way. Signals are passed between nodes along weighted connections, where the weights are the network's adjustable parameters. Basic features of neural architectures are that they are massively parallel, the weights have to be learned from a set of examples and can be adapted. Although ANNs can perform human brain-like tasks such as object and pattern recognition, speech recognition or associative memory, there is still a huge gap between biological and artificial neural nets.



(a) Schematic diagram of biological neuron.



(b) Schematic diagram of a McCulloch and Pitts Neuron.

Figure [1.1] Correspondence between a biological neuron and an artificial neuron.

The arrangement of the network's nodes and connections defines its architecture and there are many possible variations. The learning rules used to train the networks can generally be classified as supervised or unsupervised. Supervised learning rules require the desired network output to be available and they adapt the weights so that the output error is reduced. Unsupervised learning is used to organize the network's structure based only on the training inputs presented to the networks. ANNs offer solutions to problems that are very difficult to solve using traditional techniques and the potential benefits of neural approach for modeling and control are [3]:

BENEFITS OF ANNS FOR MODELING AND CONTROL

- ANNs are universal approximators: It is proven that any continuous nonlinear function can be approximated arbitrarily well over a compact interval by a multi-layer neural network that consists of one or more hidden layers.
- Parallel distributed processing: The network has a highly parallel structure and consists of many processing elements with a very simple structure, which is interesting from the viewpoint of implementation.
- Hardware implementation: Dedicated hardware is possible, resulting in additional speed.
- Learning and adaptation: The intelligence of neural networks comes from their generalization ability with respect to fresh, unknown data. Also on-line adaptation of the weights is possible.
- Multivariable systems: ANNs have naturally many inputs and outputs, which makes it easy to model multivariable systems.

Indeed ANNs have provided good solutions to many problems in various fields: examples include classification problems, vision, speech, signal processing, time series prediction, robotics, optimization, expert systems and financial applications.

1.3 Fuzzy Logic (FL)

FUZZY CONCEPTS

Fuzzy sets, introduced by Zadeh in (1965) [4] as a mathematical way to represent vagueness in linguistics, can be considered a generalization of classical set theory. The basic idea of fuzzy sets is quite easy to grasp. In a classical (non-fuzzy) set, an element of the universe either belongs to or does not belong to the set. That is, the membership of an element is *crisp*, it is either yes (in the set) or no (not in the set). A fuzzy set is a generalization of an ordinary set in that it allows the degree of membership for each element to range over the unit interval $\{0,1\}$. Thus the membership function of a fuzzy set maps each element of a universe of

discourse to its range space which, in most cases, is set to the unit interval. One of the biggest differences between crisp and fuzzy sets is that the former always has *unique* membership functions, whereas every fuzzy set has an *infinite* number of membership functions that may represent it. This enables fuzzy systems to be adjusted for maximum utility in a given situation.

In a broad sense, as pointed out by Lotfi Zadeh, any field can be fuzzified and hence generalized by replacing the concept of a crisp set in the target field by the concept of a fuzzy set. For instance, we can fuzzify some applied fields such as neural networks, genetic algorithms, stability theory, and pattern recognition to obtain fuzzy neural networks, fuzzy genetic algorithms, fuzzy stability theory and fuzzy pattern recognition respectively. The benefits of such fuzzification include greater generality, higher expressive power, an enhanced ability to model real-world problems, and a methodology for exploiting the tolerance for imprecision. Hence fuzzy logic can help to achieve tractability, robustness, and lower solution cost.

TYPICAL FUZZY FEATURES

One major feature of fuzzy logic is its ability to express the amount of ambiguity in human thinking and subjectivity (including natural language) in comparatively undistorted manner.

WHEN TO USE FUZZY LOGIC? [5]

- When the process is concerned with continuous phenomena that are not easily broken down into discrete segments;
- When a mathematical model of a process does not exist, or exists but is too difficult to encode, or is too complex to be evaluated fast enough for real-time operation, or involves too much memory on the designated chip architecture;
- When high ambient noise levels must be dealt with or it is important to use inexpensive sensors;
- When the process involves human interaction;
- And when an expert is available who can specify the rules underlying the system behavior and the fuzzy sets that represent the characteristics of each variable.

1.4 Fuzzy-Neural Integrated Systems

Fuzzy logic and Neural Networks are complementary technologies in the design of intelligent systems. Each method has merits and demerits. A comparison of these techniques is presented in Table [1.2]. To combine their merits and overcome their demerits, some integration and synthesis techniques have been proposed [5].

	FS	ANN
Mathematical model	SG	B
Learning ability	B	G
Knowledge representation	G	B
Expert knowledge	G	B
Nonlinearity	G	G
Optimization ability	B	SG
Fault tolerance	G	G
Uncertainty tolerance	G	G
Real-time operation	G	SG

The fuzzy terms used for grading are good (G), slightly good (SG), slightly bad (SB) and bad (B).

Table [1.2] Comparison of Fuzzy Systems (FS) with Artificial Neural Networks (ANN)

ANNs are essentially low-level computational structures and algorithms that offer good performance in dealing with sensory data, while fuzzy logic techniques often deal with issues such as reasoning on higher level than ANNs. However, since fuzzy systems do not have much learning capability, it is difficult for a human operator to tune fuzzy rules and membership functions from the training data set.

Since the internal layers of neural networks are always opaque to the user, the mapping rules in the network are not visible and are difficult to understand. Thus, a promising approach for reaping the benefits of both fuzzy systems and neural networks (and solving their respective problems) is to merge or fuse them into an integrated system. This fusion of two different technologies can be realized in three directions, resulting in systems with different characteristics:

1. Neural fuzzy systems: use of neural networks as tools in fuzzy models.
2. Fuzzy neural networks: fuzzification of conventional neural network models.
3. Fuzzy–neural hybrid systems: incorporation of fuzzy logic technology and neural networks into hybrid systems.

The first two systems represent supportive combinations, where one technology assists the other, and the third system exhibits collaborative combination, where two technologies are incorporated intimately to perform a common task.

Neural fuzzy systems, or Neurofuzzy systems, aim at providing fuzzy systems with automatic tuning abilities. With this approach, we witness the use of neural networks in learning or tuning membership functions and fuzzy rules of fuzzy systems. Neural network learning techniques can thus substantially reduce development time and cost while improving the performance of fuzzy systems. After learning, the user can understand the acquired rules in the network. With respect to learning speed, Neurofuzzy systems are usually faster than conventional neural networks.

1.5 Neurofuzzy for Modeling and Control

Fuzzy modeling concerns the methods of describing the characteristics of a system using fuzzy inference rules. Fuzzy modeling methods have a distinguishing feature in that they can express complex nonlinear systems linguistically.

Applications of fuzzy systems to automatic control was first reported by Mamadani [6], where, based on Zadeh's proposition, a fuzzy logic controller (FLC) was used to emulate a human operator's control of a steam engine and boiler combination. Since then, fuzzy logic control has gradually been recognized as the most significant and fruitful application for fuzzy logic and fuzzy set theory. In the past few years, advances in microprocessors and hardware technologies have created an even more diversified application domain for fuzzy logic controllers, which ranges from consumer electronics to the automobile industry. Indeed, for complex and/or ill-defined systems that are not easily subjected to conventional automatic control methods, FLCs provide a feasible alternative since they can capture the approximate, qualitative aspects of human reasoning and decision-making processes. However, without adaptive capability, the performance of FLCs relies exclusively on two factors: the availability of human experts, and the knowledge acquisition techniques to convert human expertise into appropriate fuzzy IF-THEN rules and membership functions. These two factors substantially restrict the application domain of FLCs.

In order to develop methods to incorporate human knowledge, the first issue to be considered is the types of human knowledge that are going to be used. From a high-level conceptual point of view, any control system consists of at least a plant and a controller. Therefore, human knowledge about a control system can be classified into two categories: plant knowledge and control knowledge. Depending upon the human knowledge used and the structure of the fuzzy controller, adaptive fuzzy control is classified into the following three categories:

- Indirect adaptive fuzzy control: the fuzzy controller comprises a number of fuzzy systems constructed (initially) from the plant knowledge.

- Direct adaptive fuzzy control: The fuzzy controller is a single fuzzy system constructed (initially) from the control knowledge.
- Combined indirect/direct fuzzy control: The fuzzy controller is a weighted average of the indirect and direct adaptive fuzzy controllers, therefore, both plant and control knowledge is used.

1.6 Classes of Intelligent Control

There are two distinct approaches which have been formulated in the adaptive control field: *direct* and *indirect* schemes [1]. A direct adaptive control scheme builds an explicit model of the desired controller, whereas an indirect scheme produces a model of the plant and synthesizes the control law, using a predefined optimization/inversion calculation. For instance, the majority of neurofuzzy controllers have been direct, as a fuzzy rule base is used as a controller and there exists a performance index which relates errors in the plant's output to errors in the control signal in order to update the rule base. In contrast, most of the adaptive neurocontrol schemes have been indirect, as an explicit plant model is generally constructed, to be used in a predictive control algorithm for example.

DIRECT STABILIZING CONTROLLERS

One of the simplest direct learning control schemes is shown in Figure [1.2], where a fixed, stabilizing, linear, feedback controller is used to train a learning network. The linear controller is designed so that the closed-loop system is stable in every operating region and the control signal provides a training signal for the learning module. The performance of the closed loop system depends on the current operating point, although the iterative training of the network gradually improves its performance (and the performance of the control signal) on line. As the operating point changes, the learning controller builds up a nonlinear model of the desired control surface, such that when the plant returns to the original operating point, the learnt response has not been forgotten and it can be improved upon. This requires a learning module which is temporally stable; learning about one area in the input space affects the knowledge stored in a different region minimally.

Despite the algorithm's simplicity, this approach has one main drawback; the design of the fixed linear controller. It has been claimed that the algorithm is robust with respect to the design of this controller, although the rate of convergence of the learning module depends on the quality of the training signal. A learning module is slow to adapt when the linear controller is performing poorly.

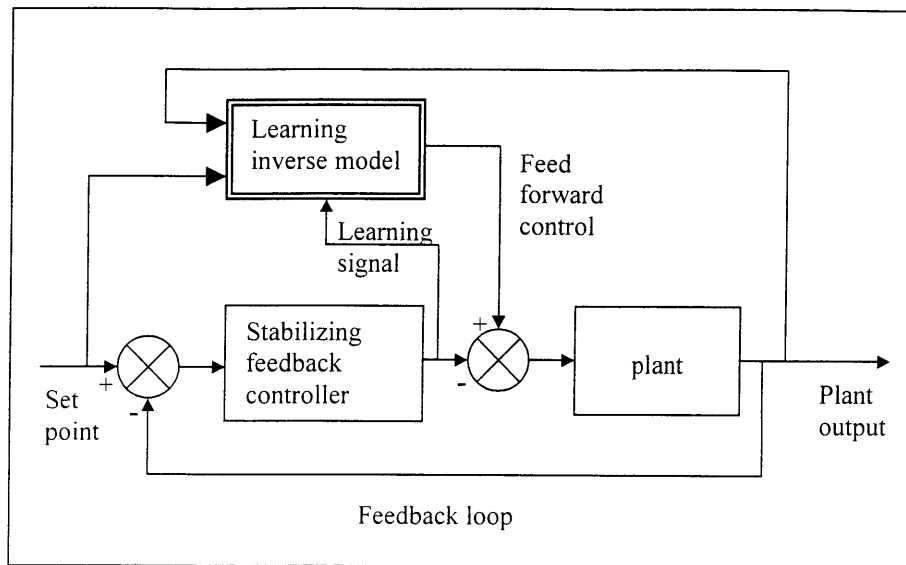


Figure [1.2] A direct learning controller; the fixed linear feedback controller is used to stabilize the system and to provide a training signal for the learning

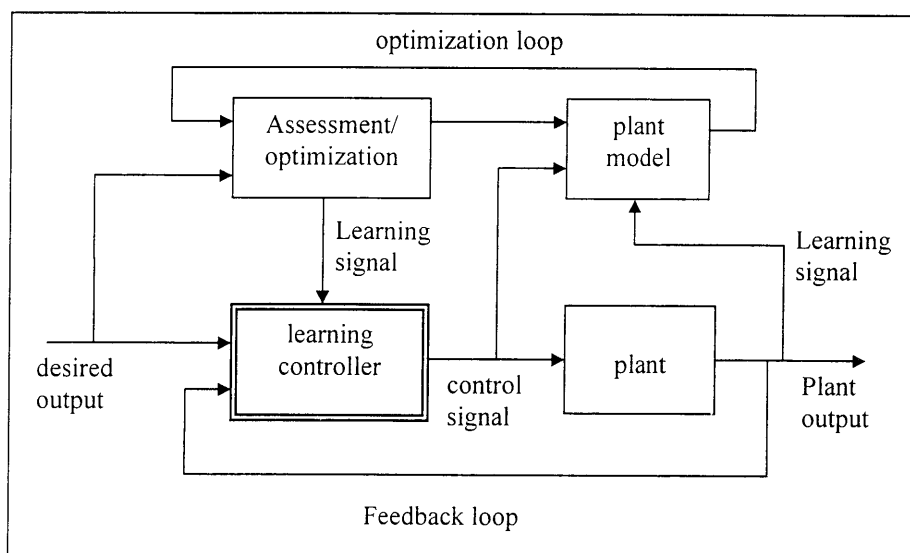


Figure [1.3] A learning predictive control architecture.

PREDICTIVE LEARNING CONTROL SCHEMES

Indirect predictive learning control schemes attempt to formulate a control strategy by assessing the effect of their actions for many time-steps into the future and selecting the current “optimal” control action, which is then applied to the plant. The architecture requires the development of a plant model, a performance function to evaluate the effect of a control action. This is illustrated in Figure [1.3], where a learning control element has also been included, so that after sufficient training, the full optimization calculation does not need to be calculated and the computing resources can be allocated to other tasks. If the plant is time-varying, the model is generally adaptive, although the initial optimization calculations may give very poor closed-loop control if the process model is poor.

When the plant model is good and the performance function and the search strategy are appropriately chosen, this control scheme can provide excellent closed-loop control. However, the multi-step ahead optimization calculation is generally very expensive and is only applied to systems which are not time-critical. Many simplifications of the above architecture can be performed which makes this technique more suitable for real-time control tasks.

MODEL REFERENCE ADAPTIVE CONTROL

The model reference learning control architecture has been widely used in the linear adaptive control field, and is shown in Figure [1.4]. The control objective is to adjust the control signal in a stable manner so that the plant’s output, $\mathbf{y}^p(t)$, asymptotically tracks the reference model’s output, $\mathbf{y}^r(t)$, i.e.:

$$\lim_{t \rightarrow \infty} \|\mathbf{y}^r(t) - \mathbf{y}^p(t)\| \leq \varepsilon \quad (1.1)$$

where ε is a small positive constant. The performance of this algorithm depends on the choice of suitable reference model and the derivation of an appropriate learning mechanism. Researchers in the sixties found that simple gradient-based learning rules were sometimes insufficient and there is no reason why this should not also be the case for more general nonlinear plant models and controllers.

INTERNAL MODEL CONTROL

Internal model control uses a similar structure to the predictive learning control scheme as shown in Figure [1.5]. A learning module is used to model the process directly, receiving the applied control signal rather than the reference signal which is used in the model reference adaptive control scheme. The error between the model and the measured plant output is used as a feedback signal and this is passed to the controller. The internal model controller is

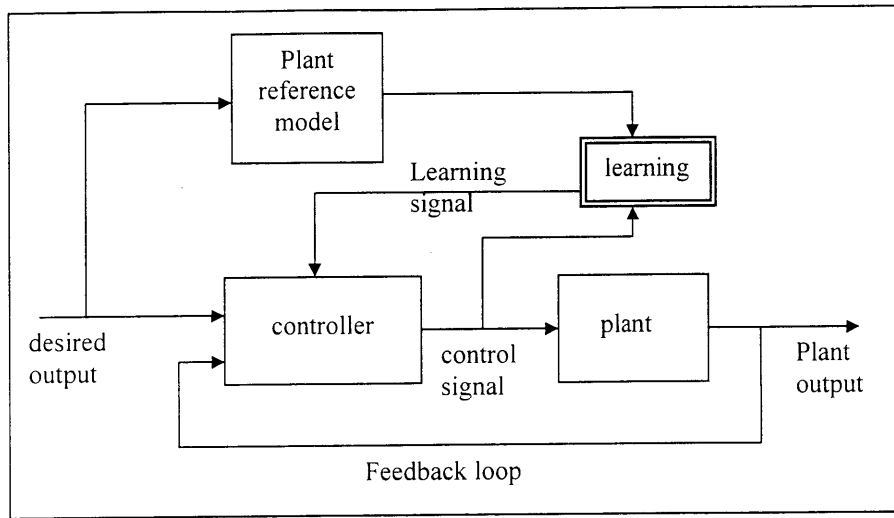


Figure [1.4] A model reference control architecture.

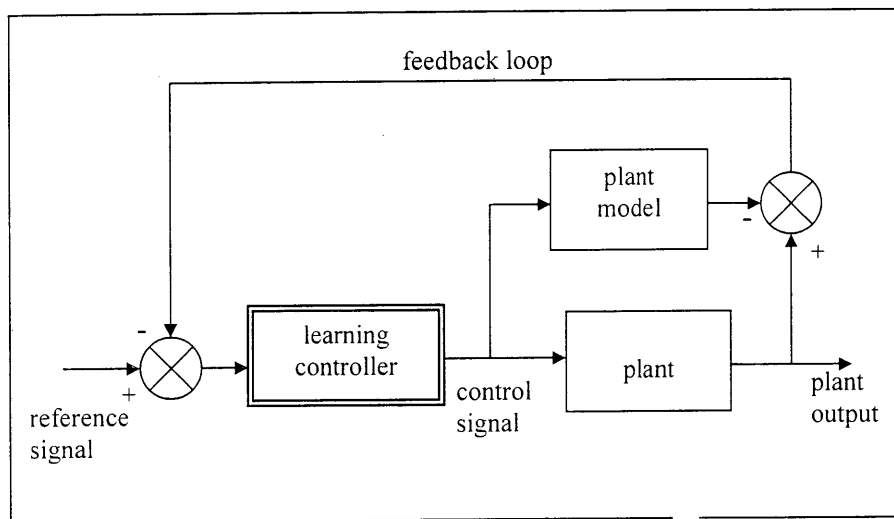


Figure [1.5] An internal model control architecture.

generally designed to be an inverse plant model (when it exists). It is claimed that this approach extends readily to nonlinear systems and yields to robustness and stability analysis.

FUZZY GAIN-SCHEDULING

One more type of controller is Fuzzy Gain-Scheduling that can capture nonlinearities or parameter dependencies on slowly-varying scheduling variables; typically, local linear controllers are interpolated by means of fuzzy logic. Gain-Scheduling is a widely used technique for controlling certain classes of nonlinear or linear time-varying systems. Rather than seeking a single robust Linear Time Invariant LTI controller for the entire operating range, gain-scheduling consists in designing an LTI controller for each operating point and in switching controller when the operating conditions change.

The common drawback of the inverse and predictive control methods is the lack of techniques for analyzing the stability, robustness and performance properties of the closed loop system. A well-known limitation of the gain-scheduling methods is that only slowly-varying trajectories are admissible for the controller to work as desired in the closed-loop [7].

Having discussed the classes of controllers that can be incorporated with fuzzy logic and neural networks, we shall now concentrate on one aspect of such controllers which is the robust fuzzy control methodology, as will be presented in the coming sections.

1.7 Robust Control

Robustness of control systems to disturbances and uncertainties has always been the central issue in feedback control. Feedback would not be needed for most control systems if there were no disturbances and uncertainties. Developing multivariable robust control methods has been the focal points in the last two decades in the control community. The state-of-the art H^∞ robust control theory is the results of this effort. The H^∞ norm was found to be appropriate for specifying both the level of plant uncertainty and the signal gain from disturbance inputs to error outputs in the controlled system.

If a control system is considered with sources of: uncertainties, noise and disturbances, two types of problem are considered.

Analysis problems: Given a controller, determine if the controlled signals (including tracking errors, control signals, etc.) satisfy the desired properties for all admissible noises, disturbances, and model uncertainties.

Synthesis problems: Design a controller in such a way that the controlled signals satisfy the desired properties for all admissible noises, disturbances and model uncertainties.

Some robust control essentials that have relations to the thesis, will be discussed in chapter two.

1.8 Robustness Behavior of Fuzzy Controllers

Control design of nonlinear systems is a difficult process, and in practical control systems, plants are always nonlinear. Thus many nonlinear control methods have been developed for nonlinear systems to overcome the difficulty in controller design for real systems. However, in these control system designs, nonlinear systems must have some predictable behaviors. For instance, the system must be minimum phase, it must be sufficiently smooth, and the parameters must be exactly known in order for the feedback linearization method to be applied. Furthermore, these control schemes are so complicated that they are not suitable for practical application.

In recent years, engineers successfully utilized fuzzy logic in a variety of industrial control applications. Despite the success, it has become evident that many basic issues remain to be addressed. The most important issue for fuzzy control systems is how to get a system design with a guarantee of *stability*, *control performance* and *robust behavior*.

1.8.1 Literature Review in Robust Fuzzy Control

STABILITY ISSUES

In fact, most of the critical comments on fuzzy control are due to the lack of a general method for its *stability analysis*. Researchers are still seeking an appropriate tool for the stability analysis of fuzzy control systems, although this situation is now being improved, as will be discussed later.

The success of fuzzy control, however, does not imply that we do not need a stability theory for it. The main drawback of the lack of stability analysis would be that we couldn't take a model-based approach to fuzzy controller design, since the stability theory will certainly give us a wider view on the future development of fuzzy control.

TAKAGI-SUGENO FUZZY MODEL

In general, fuzzy control systems can be classified as linguistic (Mamdani [8]) type and Takagi-Sugeno T-S type [9]. The linguistic type fuzzy control system is well recognized and received by the control society. The T-S type fuzzy system, which will be used in this thesis mainly focuses on the modeling aspect. It has been reported that a T-S fuzzy system can exactly model any nonlinear system [10]. On the other hand there is a main drawback of the

linguistic model compared with the T-S model in that there is a difficulty in dealing with a multidimensional system since a large number of fuzzy rules have to be used. It is, however, known that we drastically reduce the number of rules if we use T-S type for the fuzzy modeling of higher order systems. In connection with linear systems, T-S system becomes linear once the number of rules is restricted to one.

In the conventional control theory, stability analysis of a control system is based on that of an unforced dynamical system, i.e., a system without control. Therefore, needless to say, what is first needed is a methodology of the stability analysis of an unforced fuzzy dynamical system.

TAKAGI-SUGENO MODEL STABILITY

Stability of an unforced system of T-S type, *where* $u=0$ was first analyzed for a discrete case in [11] where a conventional Lyapunov function approach was taken. Related works on the stability concerns both discrete-time systems and continuous time systems. The idea in the stability analysis is to regard fuzziness in a T-S system, i.e., its nonlinearity, as uncertainty in a polytopic linear system. Hence to embed the stability problem in the control system as compatible with the conventional control theory.

There are some works in the literature that are mainly concerned with the stability analysis of a T-S fuzzy model. Interest in the stability research is now shifting from the linguistic type to the T-S type. The existence of a proper T-S fuzzy model is first assumed. In other studies, a nonlinear plant was approximated by a T-S fuzzy linear model, and then a model-based fuzzy control was developed to stabilize the T-S fuzzy linear model. The appeal of the T-S model is that the stability and performance characteristics of the system can be analyzed using a Lyapunov approach as were investigated by: [12], [13], [14], [15], [16] and [17].

In practice, the effect of approximation error between the nonlinear system and its fuzzy model does deteriorate the *stability* and *control performance* of the nonlinear system. Therefore, the stability and control performance of nonlinear control systems still need further study. M. Sugeno [18] summarizes all attempts done in the stability of both fuzzy types.

RESEARCH EFFORTS IN T-S STABILITY; THE LYAPUNOV APPROACH

Based on the stability conditions, model-based control of T-S systems has been developed for the discrete case [11], [15], [17] in addition to the continuous case [19], [20]. In these studies, state-space models are used. Tanaka and Sugeno [15] have provided a sufficient condition for the asymptotic stability of a fuzzy system in the sense of Lyapunov through the existence of a common Lyapunov function for all the subsystems. Tanaka and Sano [17] have extended this

to robust stability in case of systems with premise-parameter uncertainty. Tanaka *et al.* [13] suggest the idea of using linear matrix inequalities LMI for finding the common Lyapunov P matrix where in [12] an iterative algorithm for the choice of such common P matrix is proposed.

T-S CONTROL ISSUES

A further and a significant step has also been taken to utilize Lyapunov function based control design techniques to the control synthesis problem for T-S models. The so-called Parallel Distributed Compensation (PDC) (J. Li *et al.* [21], [22], [23], D. Niemann *et al.* [19], H. Wang *et al.* [24]) is one such control design framework that has been proposed and developed over the last few years. It has been shown that within the framework of T-S fuzzy model and PDC control design, design conditions for the stability and performance of a system can be stated in terms of the feasibility of a set of linear matrix inequalities (LMIs) (J. Li *et al.* [21], [22], D. Niemann *et al.* [19], H. Wang *et al.* [24]). This is a significant finding in the sense that there exist very efficient numerical algorithms for determining the feasibility of LMIs, so even large-scale analysis and design problems are computationally tractable. The PDC control structure utilizes a nonlinear state feedback controller which mirrors the structure of the associated T-S model. The gains of the controller can be determined automatically using an LMI formulation.

FUZZY H^∞ CONTROL

Recently, nonlinear H^∞ control schemes have been introduced to deal with the robust performance design problem of nonlinear systems. In the conventional optimal control, the plant model must be known beforehand. However, the designer has to solve a Hamilton-Jacobi equation, which is a nonlinear partial differential equation [25]. Only some very special nonlinear systems have a closed-form solution. In general, conventional nonlinear H^∞ control schemes are not suitable for practical control system design.

For practical control design, a simple fuzzy control design with guaranteed control performance is more appealing for nonlinear systems. In this work, the fuzzy linear T-S model is used to approximate a nonlinear system. Hence, a hybrid fuzzy controller is introduced to stabilize the nonlinear system, and at the same time to eliminate effects of external disturbance below a prescribed level, so that a desired H^∞ control performance can be guaranteed, B. Chen [26]. In this approach only a linear fuzzy control design is used, without complicated feedback linearization or parameter update law, although, the same H^∞ control performance is achieved. In addition, fuzzy control design with robustness

optimization is also introduced based on the optimal H^∞ control scheme by minimizing the attenuation level.

The methods attempt to combine the fuzzy linear model and H^∞ performance to obtain a simple but practical algorithm for robust performance control design of nonlinear systems. This attempt is made to create a bridge between two important control design techniques, i.e., robust and fuzzy control design, so as to supply H^∞ control design with more intelligence and fuzzy control design with better performance respectively (B. Chen *et al.* [14], B. Chen [26], K. Tanaka *et al.* [13]).

The perfect matching of parameters in an adaptive fuzzy logic system is generally deemed impossible. Therefore a desired tracking performance cannot be guaranteed in the conventional adaptive fuzzy control systems. The influence of both the fuzzy logic approximation error and external disturbance on the tracking error is attenuated to a prescribed level.

LMI TECHNIQUE

In recent years, LMIs have emerged as powerful mathematical tools for the control engineers. However, many researchers expressed their results in LMI conditions without realizing that these results can be (or have been) solved through Algebraic Riccati Equation (ARE) approach. Conversely, many classical results expressed in AREs can also be written in the form of LMIs. Within the development of fast optimization algorithms for LMIs, these classical results are more easily solved through their counterparts expressed in LMIs. For instance, J. Li *et al.* [28] shows the relation between LMIs and AREs through absolute stability criteria, robustness analysis and optimal control. This analysis of robust stability of fuzzy control systems via quadratic stabilization, H^∞ control theory is solved by LMI techniques. This LMI based technique is not only an analysis tool; it can also be utilized to automate the controller design process, LMI is a very efficient numerical algorithms and can involve large scale systems.

FUZZY GAIN-SCHEDULING

On the other hand, the research in the area of *fuzzy parameter dependent Gain-Scheduling* is limited. There are a few numbers of research such as, P. Korba *et al.* [7], [27] are proposed a fuzzy controller via Gain-Scheduling method for the T-S fuzzy model based on LYAPUNOV method and LMI techniques.

1.9 Thesis Objectives

This thesis has an objective of looking in details at synthesizing an optimal H^∞ Robust Fuzzy Control for two nonlinear dynamic systems, in which the first is a SISO system and the other is a MIMO system. In addition, the thesis also presents an approach to synthesize H^∞ Fuzzy Robust Gain-Scheduling control methodology for the two nonlinear systems.

The two problems have been framed as an LMI problem formulation where modeling has been achieved via Takagi-Sugeno fuzzy modeling of nonlinear process.

1.10 Problem Formulation

Since the thesis objectives have been centered around synthesizing *linear* H^∞ fuzzy gain-scheduling for nonlinear systems, the problem has been formulated through the consideration of obtaining suitable fuzzy sub-models. That was achieved via the employment of Takagi-Sugeno fuzzy modeling technique with clustering. Hence, H^∞ robust control synthesis have been synthesized based on the formulation of a suitable Lyapunov Function for the nonlinear systems under consideration. That was formulated as Linear Matrix Inequality Problem.

Finally, H^∞ Fuzzy Gain-Scheduling was formulated by utilizing changes in the T-S fuzzy sub-models and regarding the dynamic system as a *Linear Parametric Varying* system. Hence, seeking suitable H^∞ controller through the expression of the problem in terms of LMI formulation. Gain-scheduling is then achieved once the system under consideration undergoes change in parameters.

1.11 Thesis Outline

This thesis is presented in seven chapters and organized as follows:

CHAPTER ONE: THESIS INTRODUCTION

Chapter one serves as an introduction to the whole thesis objectives; first it gives a general introduction to the historical development of artificial intelligent and its two important approaches, fuzzy logic and neural network. After that, it gives a brief idea about their applications in modeling and control. Robust fuzzy control, the main subject of this thesis is then introduced briefly, where a survey on the work done in this area is presented. Finally, this chapter introduces the thesis objectives and the problem formulation being investigated in this thesis followed by a brief content of each chapter.

CHAPTER TWO: ROBUST CONTROL ESSENTIALS AND LINEAR MATRIX INEQUALITIES FORMULATION

This chapter presents an overall formulation of the essentials of linear robust control theory. After the introduction, the chapter gives the closed loop feedback prospect and properties. The H^∞ formulation with a level of disturbance is presented via Riccati-based and LMI-based formulations. Optimal H^∞ formulation and the mixed sensitivity H^∞ are presented as well in this chapter. Then the LMI techniques are introduced with the LMI formulation of the Quadratic LYAPUNOV Function and H^∞ . Finally, model uncertainty is discussed with a description about uncertain state-space models.

CHAPTER THREE: NEUROFUZZY ARCHITECTURE AND TAKAGI-SUGENO MODEL

Presents some important concepts and topics involving fuzzy logic (fuzzy sets, rule-based fuzzy models) and neural network (multilayer perceptron, backpropagation learning algorithm), so that to introduce the neurofuzzy systems used in this work.

CHAPTER FOUR: TAKAGI-SUGENO FUZZY CLUSTERING AND MODELING OF NONLINEAR SYSTEMS

This chapter introduces the application of the neurofuzzy systems to nonlinear dynamic system modeling. It is concentrated on the modeling via Takagi-Sugeno modeling technique and the employment of fuzzy clustering to generate suitable initial membership functions. Such initial memberships are then employed to construct the suitable T-S models.

The T-S Fuzzy modeling have been applied to model two nonlinear dynamic systems. The first is a nonlinear mass-spring-damper system, whereas, the second is a nonlinear antenna system with two coupled outputs. Furthermore, the T-S fuzzy models have been validated and checked through the use of some standard model validation techniques.

CHAPTER FIVE: FUZZY H^∞ ROBUST CONTROLLER SYNTHESIS FOR NONLINEAR SYSTEM

Chapter Five presents the synthesis of fuzzy H^∞ robust controllers of nonlinear systems that have been modeled in chapter four by T-S fuzzy models. The whole chapter is considered as a mathematical formulation of the proposed controller. In this respect, more detailed description of T-S fuzzy model is presented with some related issues such as the stability of T-S models. H^∞ robust control and its LMI formulation is presented in this chapter. Finally, the proposed controller is applied to two case studies in order to verify it. Some conclusions are presented at the end of this chapter regarding the Fuzzy H^∞ Robust Controller.

CHAPTER SIX: FUZZY-BASED H^∞ ROBUST GAIN-SCHEDULED CONTROLLER

Chapter six introduces the concept of Linear Parametric Varying systems (LPV) and discusses how a T-S fuzzy model can be viewed as a LPV system. The chapter presents issues related to the design of H^∞ Fuzzy Gain-Scheduling Control, hence it extends to the design of H^∞ Gain-Scheduling for the two nonlinear systems, which have been modeled using the concept of T-S fuzzy models. The chapter then presents two simulation examples for the H^∞ Fuzzy Gain-Scheduling.

CHAPTER SEVEN: CONCLUSIONS

Finally, chapter seven presents the thesis conclusion.

Chapter Two:

Robust Control Essentials and Linear Matrix Inequities Formulation

2.2 Feedback Properties

Considering the one-degree of freedom configuration shown in Figure [2.1]. The plant $G(s)$ and controller $K(s)$ interconnection is driven by a reference command r , output disturbances d , disturbance at plant input d_i and measurement noise n . Shown also the output y to be controlled, u_p the plant input and u are the control signals.

It is convenient to define the input loop transfer matrix, L_i , and output loop transfer matrix, L_o , as

$$L_i = KG, \quad L_o = GK \quad (2.1)$$

respectively. Here, L_i is obtained by breaking the loop at the input u of the plant while L_o is obtained by breaking the loop at the output y of the plant. The input sensitivity matrix is defined as the transfer matrix from d_i to u_p as:

$$S_i = (I + L_i)^{-1}, \quad u_p = S_i d_i \quad (2.2)$$

The output sensitivity matrix S_o is defined as the transfer matrix from d to y by:

$$S_o = (I + L_o)^{-1}, \quad y = S_o d \quad (2.3)$$

The input and output complementary sensitivity matrices are hence defined as:

$$T_i = I - S_i = L_i (I + L_i)^{-1} \quad (2.4)$$

$$T_o = I - S_o = L_o (I + L_o)^{-1} \quad (2.5)$$

respectively. The word complementary is employed to signify the fact that T is the complement of S , $T = I - S$. The matrix $I + L_i$ is called the input return difference matrix and $I + L_o$ is called the output return difference matrix.

It is easy to observe that the closed-loop system, if it is internally stable, should satisfy the following transfer function conditions:

$$y = T_o (r - n) + S_o G d_i + S_o d \quad (2.6)$$

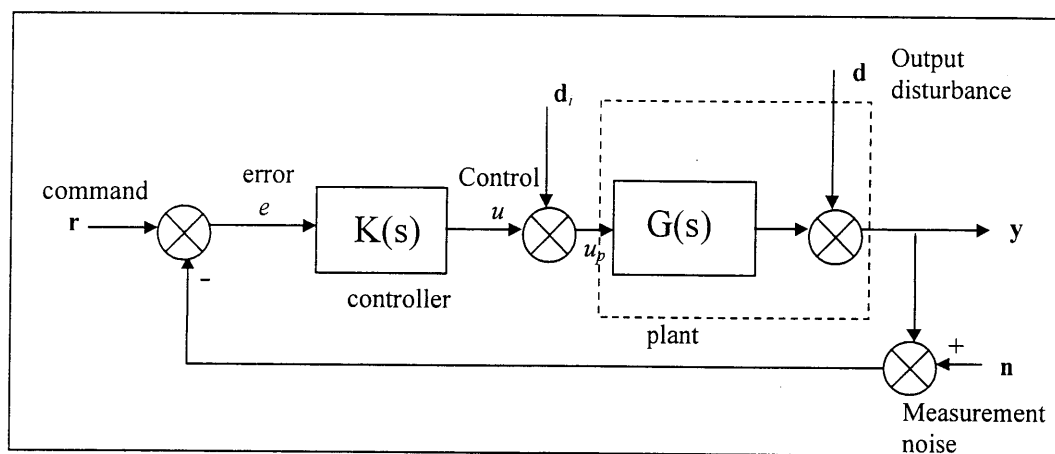


Figure [2.1] Feedback Configuration.

$$\mathbf{r} - \mathbf{y} = \mathbf{S}_o(\mathbf{r} - \mathbf{d}) + \mathbf{S}_o\mathbf{G}\mathbf{d}_i + \mathbf{T}_o\mathbf{n} \quad (2.7)$$

$$\mathbf{u} = \mathbf{K}\mathbf{S}_o(\mathbf{r} - \mathbf{n}) - \mathbf{K}\mathbf{S}_o\mathbf{d} + \mathbf{T}_i\mathbf{d}_i \quad (2.8)$$

$$\mathbf{u}_p = \mathbf{K}\mathbf{S}_o(\mathbf{r} - \mathbf{n}) - \mathbf{K}\mathbf{S}_o\mathbf{d} + \mathbf{S}_i\mathbf{d}_i \quad (2.9)$$

These four conditions represent the fundamental benefits and design objectives inherent in feedback loops. For instance, Equ. [2.6] shows that the effect of disturbance $\mathbf{d}$ on the plant output can be made “small” by making the output sensitivity function $\mathbf{S}_o$ small. Similarly, Equ. [2.9] shows that the effects of disturbance $\mathbf{d}_i$ on the plant input can be made small by making the input sensitivity function $\mathbf{S}_i$ small. The notation of smallness for a transfer matrix in a certain range of frequencies can be made explicit using frequency-dependent singular values. For instance, $\sigma_{\max}(\mathbf{S}_o) < 1$ over a frequency range would mean that the effects of disturbance $\mathbf{d}$ at the plant output are effectively desensitized over that frequency range.

Hence, a *good disturbance rejection* at the plant output $\mathbf{y}$ would require that the following conditions to be satisfied accordingly,

$$\begin{aligned} \sigma_{\max}(\mathbf{S}_o) &= \sigma_{\max}((\mathbf{I} + \mathbf{G}\mathbf{K})^{-1}) && \text{(for disturbance at plant output, } \mathbf{d}) \\ &= \frac{1}{\sigma_{\min}(\mathbf{I} + \mathbf{G}\mathbf{K})} \\ \sigma_{\max}(\mathbf{S}_o\mathbf{G}) &= \sigma_{\max}((\mathbf{I} + \mathbf{G}\mathbf{K})^{-1}\mathbf{G}) && \text{(for disturbance at plant input, } \mathbf{d}_i) \\ &= \sigma_{\max}(\mathbf{G}\mathbf{S}_i) \end{aligned}$$

be made small and good disturbance rejection at the plant input $\mathbf{u}_p$ would require that,

$$\begin{aligned} \sigma_{\max}(\mathbf{S}_i) &= \sigma_{\max}((\mathbf{I} + \mathbf{K}\mathbf{G})^{-1}) && \text{(for disturbance at plant input, } \mathbf{d}_i) \\ &= \frac{1}{\sigma_{\min}(\mathbf{I} + \mathbf{K}\mathbf{G})} \\ \sigma_{\max}(\mathbf{S}_i\mathbf{K}) &= \sigma_{\max}(\mathbf{K}(\mathbf{I} + \mathbf{K}\mathbf{G})^{-1}) && \text{(for disturbance at plant output, } \mathbf{d}) \\ &= \sigma_{\max}(\mathbf{K}\mathbf{S}_o) \end{aligned}$$

be made small, particularly in the low-frequency range where $\mathbf{d}$ and $\mathbf{d}_i$ are usually significant.

Generally, the output sensitivity $\mathbf{S}_o$ and the input sensitivity $\mathbf{S}_i$ are not equal ($\mathbf{S}_o \neq \mathbf{S}_i$) unless $\mathbf{K}$ and $\mathbf{G}$ are square and diagonal, which is true if $\mathbf{G}$ is a scalar system. Hence, small

$\sigma_{\max}(\mathbf{S}_o)$ does not necessarily imply small $\sigma_{\max}(\mathbf{S}_i)$. In other words, good disturbance rejection at the output does not necessarily mean good disturbance rejection at the plant input.

2.3 H^∞ Control Generalization

There are many ways in which feedback design problems can be cast as H^∞ optimization problems. It is very useful therefore to have a standard problem formulation into which any particular problem may be manipulated. Such a general configuration is shown in Figure [2.2]. In the figure, the plant $\mathbf{P}(s)$ and controller $\mathbf{K}(s)$ are assumed to be real, rational and proper. It will be assumed that the state-space models of $\mathbf{P}(s)$ and $\mathbf{K}(s)$ are expressed as Linear Fractional Transformation LFT and to be available where their realizations are assumed to be stabilizable and detectable.

This configuration shown in Figure [2.2] is described by,

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} = \mathbf{P}(s) \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} \quad (2.10)$$

$$\mathbf{u} = \mathbf{K}(s)\mathbf{y} \quad (2.11)$$

with state-space realization of the generalized plant $\mathbf{P}(s)$ expressed as,

$$\mathbf{P}(s) = \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{array} \right] \quad (2.12)$$

The signals are: $\mathbf{u}$ the control variables, $\mathbf{y}$ the measured variables, $\mathbf{w}$ the exogenous signals such as disturbances $\mathbf{w}_d$ and commands $\mathbf{r}$, and $\mathbf{z}$ the so-called “error” signals which are to be minimized in some sense to meet the control objectives. The closed-loop transfer function from $\mathbf{w}$ to $\mathbf{z}$ is given by LFT [29] as,

$$\mathbf{z} = F_l(\mathbf{P}, \mathbf{K})\mathbf{w} \quad (2.13)$$

where

$$F_l(\mathbf{P}, \mathbf{K}) = \mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{K}\mathbf{P}_{21} \quad (2.14)$$

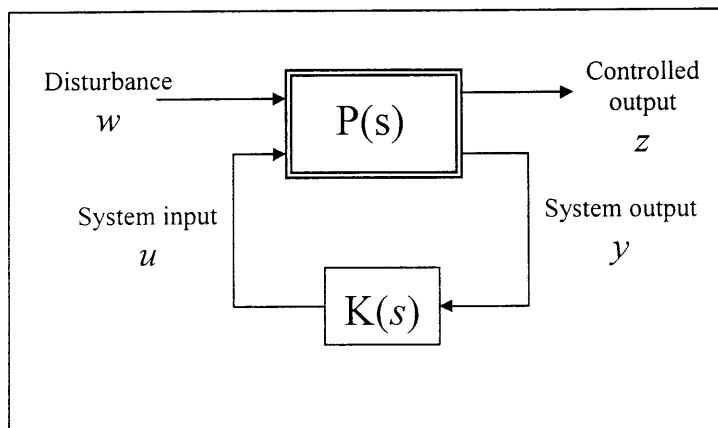


Figure [2.2] Representation of linear fractional transformation

Therefore, H^∞ control involves minimization of H^∞ norms of $F_l(\mathbf{P}, \mathbf{K})$ below a disturbance level γ .

$$\|F_l(\mathbf{P}, \mathbf{K})\|_\infty < \gamma \quad (2.15)$$

The most general, widely available and widely used algorithm for H^∞ control problems synthesis are based on the state-space solutions as given by Glover and Doyle [26].

RICCATI-BASED AND LMI-BASED APPROACHES

H^∞ can be constructed through two well-known methods, the Riccati-based and LMI-based approaches. The LMI approach is computationally more involved for large problems, in addition, LMI has the merit of eliminating the regularity restrictions attached to the Riccati-based solution. Since both approaches are state-space methods, the plant $\mathbf{P}(s)$ is given in standard LFT form as where expressed in Equ. [2.12] as,

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}_1\mathbf{w} + \mathbf{B}_2\mathbf{u} \\ \mathbf{z} &= \mathbf{C}_1\mathbf{x} + \mathbf{D}_{11}\mathbf{w} + \mathbf{D}_{12}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}_2\mathbf{x} + \mathbf{D}_{21}\mathbf{w} + \mathbf{D}_{22}\mathbf{u} \end{aligned} \quad (2.16)$$

The linear H^∞ Riccati approach is applicable to plants satisfying a number of conditions [30]. These are given as,

- $(\mathbf{A}, \mathbf{B}_2, \mathbf{C}_2)$ are stabilizable and detectable.
- Both $\mathbf{D}_{12}$ and $\mathbf{D}_{21}$ matrices have full rank.
- $\begin{bmatrix} \mathbf{A} - j\omega\mathbf{I} & \mathbf{B}_2 \\ \mathbf{C}_1 & \mathbf{D}_{12} \end{bmatrix}$ has full column rank for all ω .
- $\begin{bmatrix} \mathbf{A} - j\omega\mathbf{I} & \mathbf{B}_1 \\ \mathbf{C}_2 & \mathbf{D}_{21} \end{bmatrix}$ has full row rank for all ω .
- $\mathbf{D}_{11}=0$ and $\mathbf{D}_{22}=0$.
- $\mathbf{D}_{12} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}$ and $\mathbf{D}_{21} = \begin{bmatrix} 0 & \mathbf{I} \end{bmatrix}$
- $\mathbf{D}_{12}^T \mathbf{C}_1 = 0$ and $\mathbf{B}_1 \mathbf{D}_{21}^T = 0$
- $(\mathbf{A}, \mathbf{B}_1)$ are stabilizable and $(\mathbf{A}, \mathbf{C}_1)$ are detectable.

The H^∞ algorithms also find a sub-optimal controller $\mathbf{K}$, in such away that for a specified disturbance level γ , a stabilizing controller is found for which $\|F_l(\mathbf{P}, \mathbf{K})\|_\infty < \gamma$. If an optimal controller is required then the algorithm can be used iteratively, while reducing γ until the minimum is reached within a defined tolerance.

H^∞ Norms:

Letting a transfer function of a system given as $\mathbf{G}(s) \in RL^\infty$ and recalling that L^∞ norm of a matrix rational transfer function $\mathbf{G}$ is given by,

$$\|\mathbf{G}\|_\infty = \sup_{\omega} \sigma_{\max}(\mathbf{G}(j\omega)) \quad (2.17)$$

The interpretation of the infinity norm of a scalar transfer function $\mathbf{G}$ is *the distance in the complex plane from the origin to the farthest point on the Nyquist plot of $\mathbf{G}$* . Also it appears as the peak value on the Bode magnitude plot of $\|\mathbf{G}(j\omega)\|$.

The H^∞ norm of $\mathbf{G}(s)$ is its largest input-output RMS gain, i.e.

$$\begin{aligned} \|\mathbf{G}\|_\infty &= \sup \frac{\|\mathbf{y}\|_{L_2}}{\|\mathbf{u}\|_{L_2}} \\ \mathbf{u} &\in L_2 \\ \mathbf{u} &\neq 0 \end{aligned} \quad (2.18)$$

corresponding to the peak gain of the frequency response $\mathbf{G}(j\omega)$ i.e.

$$\|\mathbf{G}\|_\infty = \sup_{\omega} \sigma_{\max}(\mathbf{G}(j\omega)) \quad (2.19)$$

The H^∞ control synthesis is the one of disturbance rejection. It consists of minimizing the closed-loop RMS gain from $\mathbf{w}$ to $\mathbf{z}$. This is effectively minimizing the effect of the worst-case disturbance $\mathbf{w}$ on the output $\mathbf{z}$.

For a stable system given by the $\mathbf{P}(s)$ representation in Equ. [2.12], the Hamiltonian matrix is then defined as:

$$\mathbf{H} := \begin{bmatrix} \mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C} & \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ -\mathbf{C}^T[\mathbf{I} - \mathbf{D}\mathbf{D}^T]^{-1}\mathbf{C} & -(\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C}) \end{bmatrix} \quad (2.20)$$

in which $\mathbf{R} = \mathbf{I} - \mathbf{D}^T\mathbf{D}$.

Once $\|\mathbf{P}\|_\infty < 1$, $\mathbf{H}$ has no eigenvalues on the imaginary axis $\mathbf{H} \in \text{dom}(\text{Ric})$ and the stabilizing solution $\text{Ric}(\mathbf{H}) \geq 0$, where $(\text{Ric}(\mathbf{H})) > 0$ if $(\mathbf{C}, \mathbf{A})$ is observable.

2.4 Optimal H^∞ Control:

With reference to the general control configuration shown in Figure [2.2], the standard H^∞ control problem is to find all admissible controllers $\mathbf{K}$ which will minimize $F_l(\mathbf{P}, \mathbf{K})$ given by:

$$\|F_l(\mathbf{P}, \mathbf{K})\|_\infty = \max_{\omega} \sigma_{\max}(F_l(\mathbf{P}, \mathbf{K})(j\omega)) \quad (2.21)$$

Here the H^∞ norm has several interpretations in term of performance. One is that it minimizes the peak of the maximum singular value of $F_l(\mathbf{P}(j\omega), \mathbf{K}(j\omega))$. It also has a time domain interpretation as the induced (worst-case) 2-norm. Letting $\mathbf{z} = F_l(\mathbf{P}, \mathbf{K})\mathbf{w}$, then

$$\|F_l(\mathbf{P}, \mathbf{K})\|_\infty = \max_{\mathbf{w}(t) \neq 0} \frac{\|\mathbf{z}(t)\|_2}{\|\mathbf{w}(t)\|_2} \quad (2.22)$$

in which,

$$\|\mathbf{z}(t)\|_2 = \sqrt{\int_0^\infty \sum_i |z_i(t)|^2 dt} \quad \text{is the 2-norm of the vector signal.}$$

In practice, it is usually not necessary to obtain an optimal controller for the H^∞ problem, hence it is often computationally and theoretically simpler to design a sub-optimal one (*i.e.* close to the optimal ones in the sense of the H^∞ norm). Letting $\gamma_{\min}$ be the minimum value of $\|F_l(\mathbf{P}, \mathbf{K})\|_\infty$ over all stabilizing controllers $\mathbf{K}$, then the H^∞ sub-optimal control problem is: given a $\gamma < \gamma_{\min}$, find all stabilizing controllers $\mathbf{K}$ such that

$$\|F_l(\mathbf{P}, \mathbf{K})\|_\infty < \gamma \quad (2.23)$$

This can be solved by reducing γ iteratively, until an optimal solution is approached. The algorithm is summarized below with all simplifying assumptions.

2.5 Riccati-based Optimal H^∞ Algorithm

For the general control configuration of Figure [2.2] described by Eqs. [2.10]-[2.12], there exist a stabilizing controller $\mathbf{K}(s)$ such that $\|F_l(\mathbf{P}, \mathbf{K})\|_\infty < \gamma$ if and only if [31],

(i) $\mathbf{X}^\infty \geq 0$ is a solution to the algebraic Riccati equation

$$\mathbf{A}^T \mathbf{X}^\infty + \mathbf{X}^\infty \mathbf{A} + \mathbf{C}_1^T \mathbf{C}_1 + \mathbf{X}^\infty (\gamma^{-2} \mathbf{B}_1 \mathbf{B}_1^T - \mathbf{B}_2 \mathbf{B}_2^T) \mathbf{X}^\infty = 0 \quad (2.24)$$

such that

$$\text{Re } \lambda_i [\mathbf{A} + (\gamma^{-2} \mathbf{B}_1 \mathbf{B}_1^T - \mathbf{B}_2 \mathbf{B}_2^T) \mathbf{X}^\infty] < 0, \quad \forall_i; \text{ and} \quad (2.25)$$

(ii) $\mathbf{Y}^\infty \geq 0$ is a solution to the algebraic Riccati equation

$$\mathbf{A} \mathbf{Y}^\infty + \mathbf{Y}^\infty \mathbf{A} + \mathbf{B}_1 \mathbf{B}_1^T + \mathbf{Y}^\infty (\gamma^{-2} \mathbf{C}_1^T \mathbf{C}_1 - \mathbf{C}_2^T \mathbf{C}_2) \mathbf{Y}^\infty = 0 \quad (2.26)$$

such that

$$\text{Re } \lambda_i [\mathbf{A} + \mathbf{Y}^\infty (\gamma^{-2} \mathbf{C}_1^T \mathbf{C}_1 - \mathbf{C}_2^T \mathbf{C}_2) \mathbf{Y}^\infty] < 0, \quad \forall_i; \text{ and} \quad (2.27)$$

(iii) $\rho(\mathbf{X}^\infty \mathbf{Y}^\infty) < \gamma^2$

All such controllers are then given by $\mathbf{K}(s) = F_l(\mathbf{K}_c(s), \mathbf{Q})$, in which

$$\mathbf{K}_c(s) = \left[\begin{array}{c|cc} \mathbf{A}^\infty & -\mathbf{Z}^\infty \mathbf{L}^\infty & \mathbf{Z}^\infty \mathbf{B}_2 \\ \hline \mathbf{F}^\infty & 0 & \mathbf{I} \\ \hline -\mathbf{C}_2 & \mathbf{I} & 0 \end{array} \right] \quad (2.28)$$

$$\mathbf{F}^\infty = -\mathbf{B}_2^T \mathbf{X}^\infty, \quad \mathbf{L}^\infty = -\mathbf{Y}^\infty \mathbf{C}_2^T, \quad \mathbf{Z}^\infty = (\mathbf{I} - \gamma^{-2} \mathbf{Y}^\infty \mathbf{X}^\infty)^{-1} \quad (2.29)$$

$$\mathbf{A}^\infty = \mathbf{A} + \gamma^{-2} \mathbf{B}_1 \mathbf{B}_1^T \mathbf{X}^\infty + \mathbf{B}_2 \mathbf{F}^\infty + \mathbf{Z}^\infty \mathbf{L}^\infty \mathbf{C}_2 \quad (2.30)$$

and $\mathbf{Q}(s)$ is any stable proper transfer function such that $\|\mathbf{Q}\|_\infty < \gamma$. For $\mathbf{Q}(s)=0$, we get

$$\mathbf{K}(s) = \mathbf{K}_{cl1}(s) \quad (2.31)$$

$$= -\mathbf{Z}^\infty \mathbf{L}^\infty (s\mathbf{I} - \mathbf{A}^\infty)^{-1} \mathbf{F}^\infty \quad (2.32)$$

This is called the central controller and has the same number of states as the generalized plant $\mathbf{P}(s)$.

γ -ITERATION: If we desire a controller that achieves $\gamma_{\min}$, to within a specified tolerance, then a bisection can be performed on γ until its value is sufficiently accurate. The above result provides a test for each value of γ to determine whether it is less than $\gamma_{\min}$ or greater than $\gamma_{\min}$.

2.6 H^∞ Mixed-Sensitivity Problems

A good performance is required in some frequency range, typically some low-frequency range $(0, \omega_l)$ [31],

$$\sigma_{\min}(\mathbf{GK}) \ll 1, \quad \sigma_{\min}(\mathbf{KG}) \ll 1, \quad \sigma_{\min}(\mathbf{K}) \leq M, \quad (2.33)$$

where M is a bounded value. A good robustness and good sensor noise rejection required in some frequency range, typically some high-frequency range (ω_h, ∞) ,

$$\sigma_{\max}(\mathbf{GK}) \ll 1, \quad \sigma_{\max}(\mathbf{KG}) \ll 1, \quad \sigma_{\max}(\mathbf{K}) \leq M, \quad (2.34)$$

where M is not too large. Such design requirements are shown graphically in Figure [2.3].

WEIGHTED H^∞ PERFORMANCE

In this sub-section, we consider how to formulate some performance objectives into mathematically tractable problems. The performance objectives of a feedback system can usually be specified in terms of requirements on the sensitivity functions and/or complementary sensitivity functions or in terms of some other closed-loop transfer functions.

For instance, the performance criteria for a scalar system may be specified as requiring,

$$\begin{cases} |\mathbf{S}(j\omega)| \leq \varepsilon, & \forall \omega \leq \omega_o, \\ |\mathbf{S}(j\omega)| \leq M, & \forall \omega > \omega_o \end{cases} \quad (2.35)$$

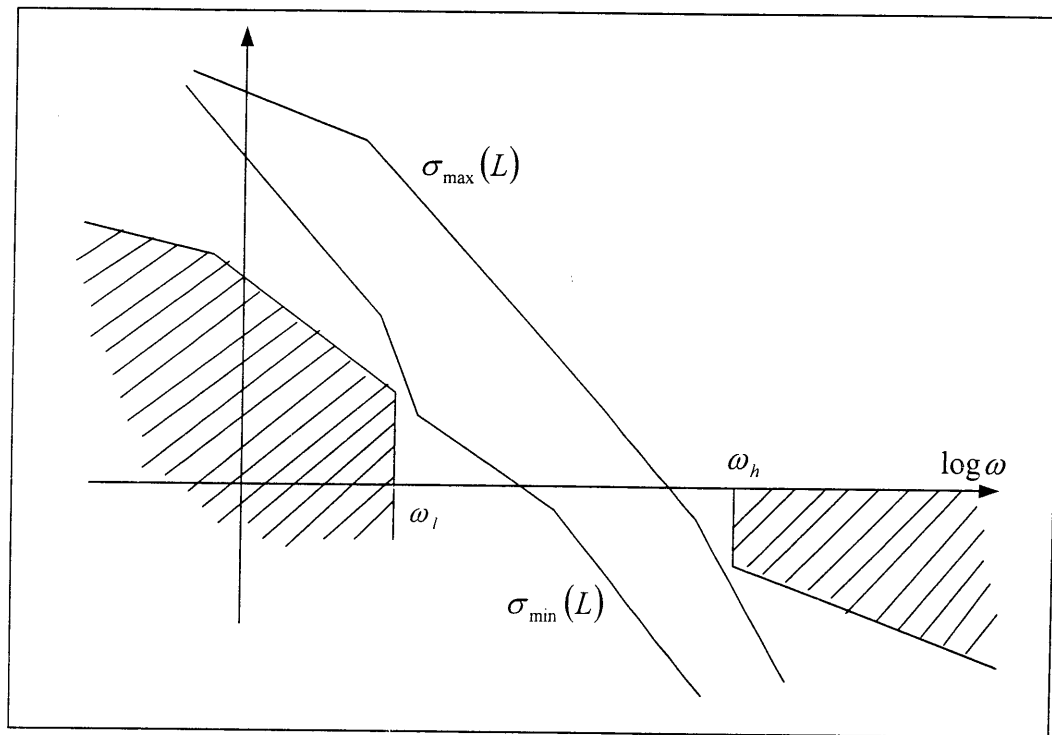


Figure [2.3] Desired loop gain

where $S(j\omega) = 1/(1 + P(j\omega)K(j\omega))$. However, it is much more convenient to reflect the system performance objectives by choosing appropriate weighting functions. For instance, the preceding objective can be expressed as,

$$|W_e(j\omega)S(j\omega)| \leq 1, \quad \forall \omega \quad (2.36)$$

with

$$|W_e(j\omega)| = \begin{cases} 1/\epsilon, & \forall \omega \leq \omega_o \\ 1/M, & \forall \omega > \omega_o \end{cases} \quad (2.37)$$

To use W_e in control design, a rational transfer function $W_e(s)$ is usually used to approximate the foregoing frequency response.

The advantage of using weighted performance specifications is obvious in multivariable system design. First, some components of a vector signal are usually more important than others. Second, each component of the signal may not be measured in the same units. Therefore, weighting functions are essential to make these components comparable. Furthermore, we might be primarily interested in rejecting errors in a certain frequency range.

In general, the standard feedback scheme in Figure [2.1] can be modified into Figure [2.4]. The weighting functions in the figure are chosen to reflect the design objectives and knowledge of the disturbances and sensor noise. For instance, W_d and W_i may be chosen to reflect the frequency contents of the disturbances $\mathbf{d}$ and $\mathbf{d}_i$, or they may be used to model the disturbance power spectrum depending on the nature of the signals involved in the practical systems.

The weighting matrix W_n is used to model the frequency contents of the sensor noise while W_e may be used to reflect the requirements on the shape of certain closed-loop transfer functions (for example, the shape of the output sensitivity function). Similarly, W_u may be used to reflect some restrictions on the control or actuator signals, and the dashed pre-compensator W_r is an optional element used to achieve deliberate command shaping or to represent a nonunity feedback system in equivalent unity feedback form.

Hence, an essential step in the controller design process is to choose the appropriate weights, W_e , W_d , W_u , and possibly W_n , W_i , W_r .

Mixed-sensitivity is the name given to transfer function shaping problems in which the sensitivity function is shaped along with one or more other closed-loop transfer functions such as $K(s)S$ or the complementary sensitivity T .

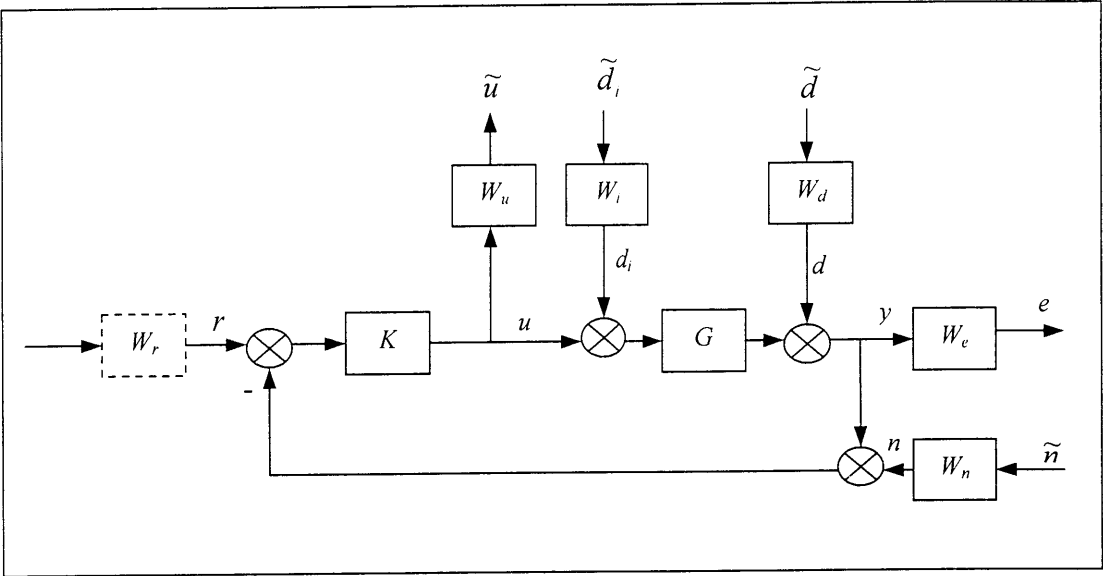


Figure [2.4] Standard feedback configuration with weights

Putting it in standard forms, the output weights are then given by minimizing the following cost function,

$$\sup_{\|\tilde{d}\|_2 \leq 1} \|\mathbf{e}\|_2 = \|\mathbf{W}_e \mathbf{S}_o \mathbf{W}_d\|_\infty \quad (2.38)$$

subject to some restriction on the control energy or control bandwidth,

$$\sup_{\|\tilde{d}\|_2 \leq 1} \|\tilde{\mathbf{u}}\|_2 = \|\mathbf{W}_u \mathbf{K} \mathbf{S}_o \mathbf{W}_d\|_\infty \quad (2.39)$$

or more frequently, by introducing a parameter ρ and a mixed criterion,

$$\sup_{\|\tilde{d}\|_2 \leq 1} \{\|\mathbf{e}\|_2^2 + \rho^2 \|\tilde{\mathbf{u}}\|_2^2\} = \left\| \begin{bmatrix} \mathbf{W}_e \mathbf{S}_o \mathbf{W}_d \\ \rho \mathbf{W}_u \mathbf{K} \mathbf{S}_o \mathbf{W}_d \end{bmatrix} \right\|_\infty^2 \quad (2.40)$$

The parameter ρ in Equ. [2.40] defines the tradeoff between good disturbance rejection at the output and control effort or disturbance and sensor noise rejection at the actuators. Alternatively, if the system robust stability margin is the major concern, the weighted complementary sensitivity has to be limited. Thus the whole cost function may be expressed as,

$$\left\| \begin{bmatrix} \mathbf{W}_e \mathbf{S}_o \mathbf{W}_d \\ \rho \mathbf{W}_1 \mathbf{T}_o \mathbf{W}_2 \end{bmatrix} \right\|_\infty \quad (2.41)$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the frequency-dependent uncertainty scaling matrices, these design problems are usually called H^∞ mixed-sensitivity problems.

2.7 Model Uncertainty

For control design purposes, the possibly complex behavior of dynamical systems must be approximated by models of relatively low complexity. The gap between models and the true physical system is called the model uncertainty. Another cause of uncertainty is the imperfect knowledge of some components of the system, or the alteration of their behavior due to changes in operating conditions, aging, etc. Finally, uncertainty also stems from physical parameters whose value is only approximately known or varies in time. Model uncertainty should be distinguished from exogenous actions such as disturbances or measurement noise.

There are two major classes of uncertainty:

- Dynamical uncertainty, which consists of dynamical components neglected in the linear model as well as of variations in the dynamical behavior during operation.

- Parameter uncertainty, which stems from imperfect knowledge of the physical parameter values, or from variations of these parameters during operation.

Other important characteristics of uncertainty include whether it is linear or nonlinear, time invariant or time varying, and whether it is parameter-dependent or not. Model uncertainty is generally a combination of dynamical and parametric uncertainty, and may arise at several different points in the control loop. For instance, there may be dynamical uncertainty on system actuators and parametric uncertainty on some sensor coefficients.

2.7.1 Uncertain State-Space Model

Physical models of a system often lead to a state-space description of its dynamical behavior. The resulting state-space equations typically involve physical parameters whose value is only approximately known, as well as approximations of complex and possibly nonlinear phenomena. In other words, the system is described by an uncertain state-space model,

$$\begin{aligned} \mathbf{E}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{aligned} \quad (2.42)$$

where the state-space matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{E}$ depends on uncertain and/or time-varying parameters or vary in some bounded sets of the space of matrices.

POLYTOPIC MODELS

We call a polytopic system a linear time-varying system,

$$\begin{aligned} \mathbf{E}(t)\dot{\mathbf{x}} &= \mathbf{A}(t)\mathbf{x} + \mathbf{B}(t)\mathbf{u} \\ \mathbf{y} &= \mathbf{C}(t)\mathbf{x} + \mathbf{D}(t)\mathbf{u} \end{aligned} \quad (2.43)$$

whose system matrix $\mathbf{S}(t) = \begin{bmatrix} \mathbf{A}(t) + j\mathbf{E}(t) & \mathbf{B}(t) \\ \mathbf{C}(t) & \mathbf{D}(t) \end{bmatrix}$ varies within a fixed polytope of matrices [32], i.e.,

$$\begin{aligned} \mathbf{S}(t) &\in \text{Co}\{\mathbf{S}_1, \dots, \mathbf{S}_k\} \\ &= \left\{ \sum_{i=1}^k \alpha_i \mathbf{S}_i : \alpha_i \geq 0, \sum_{i=1}^k \alpha_i = 1 \right\} \end{aligned} \quad (2.44)$$

where $\mathbf{S}_1, \dots, \mathbf{S}_k$ are given vertex systems:

$$S_1 = \begin{bmatrix} A_1 + jE_1 & B_1 \\ C_1 & D_1 \end{bmatrix}, \dots, S_k = \begin{bmatrix} A_k + jE_k & B_k \\ C_k & D_k \end{bmatrix} \quad (2.45)$$

$S(t)$ is a convex combination of the system matrices $S_1, \dots, S_k$. The nonnegative numbers $\alpha_1, \dots, \alpha_k$ are called the polytopic coordinates of S .

AFFINE PARAMETER-DEPENDENT MODELS

The equations of physics often involve uncertain or time-varying coefficients. Once a system is linear, this gives rise to parameter-dependent models of the form,

$$\begin{aligned} E(\delta)\dot{x} &= A(\delta)x + B(\delta)u \\ y &= C(\delta)x + D(\delta)u \end{aligned} \quad (2.46)$$

where $A(\cdot), \dots, E(\cdot)$ are known functions of some parameter vector $\delta = (\delta_1, \dots, \delta_n)$.

The parameter uncertainty range can be described as a *box* in the parameter space. This corresponds to cases where each uncertain or time-varying parameter δ_i ranges between two empirically determined *extremal* values $\delta_{i\min}$ and $\delta_{i\max}$,

$$\delta_i \in [\delta_{i\min}, \delta_{i\max}] \quad (2.47)$$

If $\delta = (\delta_1, \dots, \delta_n)$ is the vector of all uncertain parameters, Equ. [2.47] delimits a hyper-rectangle of the parameter space $\mathbf{R}^n$ called the parameter box, as illustrated in Figure [2.5].

Robustness against fast parameter variations is more stringent than robustness against constant but uncertain parameters. Alternatively, uncertain parameter vectors can be specified as ranging in a polytope of the parameter space $\mathbf{R}^n$ like the one shown in Figure [2.6], for two parameters, this polytope is characterized by three vertices.

2.8 LMI Techniques

Linear Matrix Inequalities (LMIs) technique has emerged as powerful design tool in areas ranging from control engineering to system identification and structural design. These are a number of issues that make LMI techniques appealing [32]:

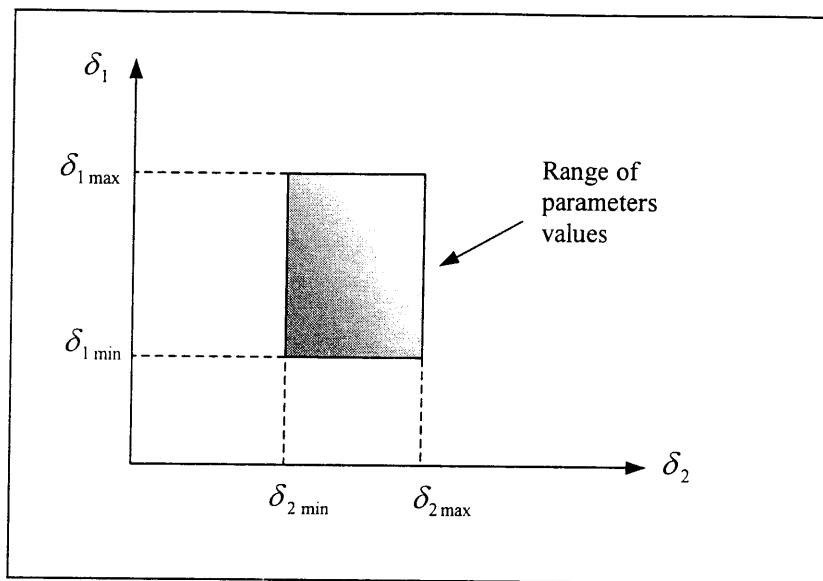


Figure [2.5] Parameter Box.

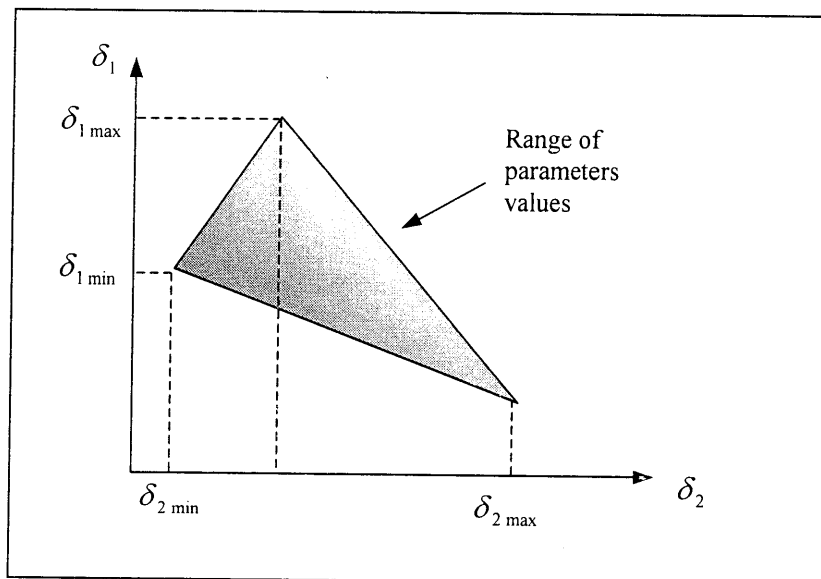


Figure [2.6] Polytopic parameter range.

- A variety of design specifications and constraints can be expressed as LMIs formulation.
- Once formulated in terms of LMIs, a problem can be solved exactly by efficient convex optimization algorithms.
- While most problems with multiple constraints or objectives lack analytical solutions in terms of matrix equations, they often remain tractable in the LMI framework. This makes LMI-based design a valuable alternative to classical analytical methods.

A Linear Matrix Inequality (LMI) is any constraint of the form,

$$\mathbf{A}(\mathbf{x}) = A_0 + x_1 A_1 + \dots x_N A_N < 0 \quad (2.48)$$

where,

- $\mathbf{x} = (x_1, \dots, x_N)$ is a vector of unknown scalars (the decision or optimization variables)
- $A_0, \dots, A_N$ are given symmetric matrices
- “ < 0 ” stands for “negative definite”, i.e., the largest eigenvalue of $\mathbf{A}(\mathbf{x})$ is negative.

The constraints $\mathbf{A}(x) > 0$ and $\mathbf{B}(x) > 0$ are special cases of Equ. [2.48] since they can be rewritten as $-\mathbf{A}(x) < 0$ and $-\mathbf{B}(x) < 0$, respectively.

- Its solution set, called the feasible set, is a convex subset of R^N .
- Finding a solution x to Equ. [2.48], if any, is a convex optimization problem.

Convexity has an important consequence: even though Equ. [2.48] has no analytical solution in general, it can be solved numerically with guarantees of finding a solution once a solution exists.

A system of LMI constraints can be regarded as a single LMI since

$$\begin{cases} A_1(x) < 0 \\ \vdots \\ A_K(x) < 0 \end{cases} \quad \text{is equivalent to} \quad \mathbf{A}(x) = \text{diag}(A_1(x), \dots, A_K(x)) < 0 \quad (2.49)$$

where $\text{diag}(A_1(x), \dots, A_K(x))$ denotes the block-diagonal matrix with $A_1(x), \dots, A_K(x)$ on its diagonal.

In most control applications, LMIs do not naturally arise in the canonical form Equ. [2.48], but rather in the form of,

$$L(X_1, \dots, X_n) < R(X_1, \dots, X_n) \quad (2.50)$$

where $L(\cdot)$ and $R(\cdot)$ are affine functions of some structured matrix variables $X_1, \dots, X_n$.

For instance, a simple example is the Lyapunov inequality

$$\mathbf{A}^T \mathbf{X} + \mathbf{X} \mathbf{A} < 0 \quad (2.51)$$

where the unknown $\mathbf{X}$ is a symmetric matrix. Defining $x_1, \dots, x_N$ as the independent scalar entries of $\mathbf{X}$, this LMI could be written in the form Equ. [2.48].

LMI USE IN CONTROL:

Many control problems and design specifications have LMI formulation. This is especially true for Lyapunov-based analysis and design, optimal LQG control, H^∞ control, covariance control, and other techniques. Further applications of LMIs arise in estimation, identification, optimal design, structural design, and so on. The main strength of LMI formulations is the ability to combine various design constraints or objectives in a numerically tractable manner.

2.8.1 LMI Formulation of Quadratic Lyapunov Functions:

In terms of time varying control systems, quadratic stability is useful for analysis, as it can be expressed as,

$$\mathbf{E}(t)\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (2.52)$$

Given such system, a sufficient condition for asymptotic stability is the existence of a positive-definite quadratic Lyapunov Function,

$$V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x} \quad (2.53)$$

in such a way, $\dot{V}(\mathbf{x}) < 0$ as given by

$$\frac{dV(\mathbf{x}(t))}{dt} < 0 \quad (2.54)$$

along all state trajectories. In terms of $\mathbf{Q} = \mathbf{P}^{-1}$, this is equivalent to

$$\mathbf{A}(t)\mathbf{Q}\mathbf{E}(t)^T + \mathbf{E}(t)\mathbf{Q}\mathbf{A}(t)^T < 0 \quad (2.55)$$

at all times t .

Assessing quadratic stability is not tractable in general, since Equ. [2.55] places an infinite number of constraints on $\mathbf{Q}$. However, Equ. [2.55] can be reduced to a finite set of LMI constraints in the following case:

$\mathbf{A}(t)$ and $\mathbf{E}(t)$ are fixed affine functions of some time-varying parameters $\delta_1(t), \dots, \delta_n(t)$,

$$\mathbf{A}(t) = \mathbf{A}_o + \delta_1(t)\mathbf{A}_1 + \dots + \delta_n(t)\mathbf{A}_n \quad (2.56)$$

$$\mathbf{E}(t) = \mathbf{E}_o + \delta_1(t)\mathbf{E}_1 + \dots + \delta_n(t)\mathbf{E}_n \quad (2.57)$$

This is referred to as an affine parameter-dependent model, which corresponds to systems whose state-space equations depend affinely on time-varying physical parameters. The quadratic Lyapunov Functions guarantee stability for arbitrarily fast time variations, since this is their main source of conservatism.

LMI FOR LINEAR TIME-VARYING SYSTEMS:

Quadratic Lyapunov functions are also useful to assess robust RMS performance. Specifically, a time-varying system

$$\begin{cases} \mathbf{E}(t)\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} + \mathbf{B}(t)\mathbf{u} \\ \mathbf{y} = \mathbf{C}(t)\mathbf{x} + \mathbf{D}(t)\mathbf{u} \end{cases} \quad (2.58)$$

with zero initial state satisfies

$$\|\mathbf{y}\|_{L_2} < \gamma \|\mathbf{u}\|_{L_2} \quad (2.59)$$

for all bounded input $\mathbf{u}(t)$ if there exists a positive-definite LYAPUNOV function $V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}$ such that,

$$\frac{dV(\mathbf{x})}{dt} + \mathbf{y}^T \mathbf{y} - \gamma^2 \mathbf{u}^T \mathbf{u} < 0 \quad (2.60)$$

In terms of $\mathbf{Q} = \mathbf{P}^{-1}$, this is equivalent to the family of inequalities given by,

$$\begin{bmatrix} \mathbf{A}(t)\mathbf{Q}\mathbf{E}(t)^T + \mathbf{E}(t)\mathbf{Q}\mathbf{A}(t)^T & \mathbf{B}(t) & \mathbf{E}(t)\mathbf{Q}\mathbf{C}(t)^T \\ \mathbf{B}(t)^T & -\gamma\mathbf{I} & \mathbf{D}(t)^T \\ \mathbf{C}(t)\mathbf{Q}\mathbf{E}(t)^T & \mathbf{D}(t) & -\gamma\mathbf{I} \end{bmatrix} < 0 \quad (2.61)$$

the smallest γ for which such a Lyapunov function exists is called the Quadratic H^∞ performance. This is an upper bound on the worst-case RMS gain of the system.

LMI FOR LINEAR PARAMETRIC VARYING SYSTEMS:

Sufficient LMI conditions for quadratic stability are defined by considering a parameter-dependent model as,

$$\mathbf{E}(\delta)\dot{\mathbf{x}} = \mathbf{A}(\delta)\mathbf{x}, \quad (2.62)$$

with,

$$\mathbf{A}(\delta) = \mathbf{A}_o + \delta_1(t)\mathbf{A}_1 + \dots + \delta_n(t)\mathbf{A}_n \quad (2.63)$$

$$\mathbf{E}(\delta) = \mathbf{E}_o + \delta_1(t)\mathbf{E}_1 + \dots + \delta_n(t)\mathbf{E}_n \quad (2.64)$$

where $\delta_i(t) \in [\delta_{i,\min}, \delta_{i,\max}]$, and let

$$\mathcal{V} = \{(\omega_1, \dots, \omega_n) : \omega_i \in [\delta_{i,\min}, \delta_{i,\max}]\} \quad (2.65)$$

denoting the set of corners of the corresponding *parameter box* as was shown in Figure [2.6].

The dynamical system is quadratically stable if there exist symmetric matrices $\mathbf{Q}$ and $\{\mathbf{M}_i\}_{i=1}^n$ such that,

$$\mathbf{A}(\omega)\mathbf{Q}\mathbf{E}(\omega)^T + \mathbf{E}(\omega)\mathbf{Q}\mathbf{A}(\omega)^T + \sum_i \omega_i^2 \mathbf{M}_i < 0 \quad \text{for all } \omega \in \mathcal{V} \quad (2.66)$$

$$\mathbf{A}_i \mathbf{P} \mathbf{E}_i^T + \mathbf{E}_i \mathbf{P} \mathbf{A}_i^T + \mathbf{M}_i \geq 0 \quad \text{for all } i=1, \dots, n \quad (2.67)$$

$$\mathbf{M}_i \geq 0$$

$$\mathbf{Q} > \mathbf{I}$$

2.8.2 H^∞ LMI Formulation

Given a state-space realization of a system $\mathbf{G}(s)$ in an LFT format as,

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}_1\mathbf{w} + \mathbf{B}_2\mathbf{u} \\ \mathbf{z} &= \mathbf{C}_1\mathbf{x} + \mathbf{D}_{11}\mathbf{w} + \mathbf{D}_{12}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}_2\mathbf{x} + \mathbf{D}_{21}\mathbf{w} + \mathbf{D}_{22}\mathbf{u} \end{aligned} \quad (2.68)$$

hence for H^∞ performance, i.e. (the closed-loop RMS gain from $\mathbf{w}$ to $\mathbf{z}$ does not exceed a γ) if and only if there exists a symmetric matrix $\mathbf{X}^\infty$ such that the following matrix inequalities are satisfied [33]:

$$\begin{bmatrix} (\mathbf{A} + \mathbf{B}_2 \mathbf{K}) \mathbf{X}^\infty + \mathbf{X}^\infty (\mathbf{A} + \mathbf{B}_2 \mathbf{K})^T & \mathbf{B}_1 & \mathbf{X}^\infty (\mathbf{C}_1 + \mathbf{D}_{12} \mathbf{K})^T \\ \mathbf{B}_1^T & -\mathbf{I} & \mathbf{D}_{11}^T \\ (\mathbf{C}_1 + \mathbf{D}_{12} \mathbf{K}) \mathbf{X}^\infty & \mathbf{D}_{11} & -\gamma^2 \mathbf{I} \end{bmatrix} < 0 \quad (2.69)$$

$$\mathbf{X}^\infty > 0$$

such formulation is a well known theory through which LMI has been employed to compute the associated suitable vector $\mathbf{X}^\infty$ to synthesis an H^∞ control structure.

2.9 Conclusions

This chapter has looked in depth at essentials of linear robust control theory and the main formula for LFT dynamic systems. In this sense, the chapter has been organized to present the main theorems which will be employed later in the forthcoming chapters. Hence we have looked at the formulation of the optimal H^∞ , Riccati-based, H^∞ - based formulation and the LMI H^∞ formulation, since they represent the mathematical bases for the synthesis of a robust H^∞ control paradigms.

In terms of defining the LMI technique, the chapter has indeed presented the LMI formulation for solving the H^∞ robust control problem once the system under concern is not easily formulated in terms of the Riccati-base formulation. Furthermore, since the H^∞ robust controller synthesis is expressed as LMI problems, this chapter has looked at the LMI theory in depth as it will be employed later in chapter five.

Of essential importance presented in this chapter the concept of the H^∞ mixed-sensitivity problem and the weighted H^∞ performance. Such performances are essential once the designer is looking at specific system behavior that is synthesized at different frequency ranges.

The chapter has looked at some theories related to the issue of model uncertainty, since a modeled system can be viewed at as a nominal model in addition to specific parametric variations in addition to fixed parameters. Such theories are of a focus, specifically once looking at the design of a fuzzy gain scheduling controller synthesis, which will be presented in chapter six.

Finally, the H^∞ design of Linear Parametric Varying Systems via the LMI formulation has been presented and discussed with the emphasis on the LFT system realization. In conclusion, the chapter has been casted to introduce the essentials of robust linear control theories in terms of the Riccati-based and LMI-based formulations, since the two approaches will be employed to synthesise the robust fuzzy control system design in chapters five and six.

Chapter Three:

Neurofuzzy Architecture and Takagi-Sugeno Fuzzy Model

3.1 Introduction

Fuzzy systems can be implemented through utilization of the neural networks learning capabilities. In this sense, such an integration results in a neurofuzzy systems. Neurofuzzy systems are a class of adaptive networks that are functionally equivalent to fuzzy inference systems. This class is referred to ANFIS, which stands for Adaptive NeuroFuzzy Inference System proposed by Jang [2].

The most popular fuzzy logic systems in the literature may be classified into two main types: fuzzy logic systems with fuzzifier and defuzzifier (linguistic model) and Takagi and Sugeno T-S fuzzy systems. Each type has its own characteristics and applications. T-S fuzzy system is a special class of neurofuzzy systems.

For modeling and control applications, the employment of neurofuzzy system for implementing the Takagi-Sugeno fuzzy model have indeed resulted in establishing a bridge between neural networks and linearized fuzzy model. Breaking a global nonlinear model into a number of linear models is one of the potentials of using T-S models. However, there are a number of issues regarding their use and training in addition to the converge toward sub-models as will be the main issues presented in this chapter.

In order to introduce the concept of ANFIS and their relation to T-S models architectures which is used in this thesis, some concepts of fuzzy logic systems and artificial neural networks are explained. First, an introductory concepts about fuzzy sets is explained in section 3.2 then the architecture of the rule-based fuzzy models are explained in section 3.3 for both types. In section 3.4 basics of ANN is introduced, that is the multilayer perceptron which will be used in the sequel for modeling and control purposes. The back-propagation learning paradigm for the multilayer perceptron is presented in section 3.4.2. ANFIS is discussed in section 3.5 which shows how to decompose the parameter set to facilitate the hybrid learning rule for ANFIS architectures representing Takagi-Sugeno fuzzy model. Finally, section 3.6 concludes some remarks of the chapter.

3.2 Fuzzy sets

Fuzzy systems are used to represent imprecision. To describe this Zadeh [4], [34] introduced the concept of a fuzzy set. Fuzzy represents vague descriptions of objects. Mathematically, a fuzzy set, A , is a function defined on the universe of discourse, U , given by,

$$\mu_A(x): U \longrightarrow [0,1] \quad (3.1)$$

where A is the *fuzzy label* or linguistic variable describing the variable x . The universe of discourse of a variable is its range and can be either continuous or discrete. As variables found in modeling and control applications are typically continuous, the former is considered here and therefore $\mu_A(x)$ is a continuous function. As an extension to Boolean logic $\mu_A(x)$ represents the grade of membership of x belonging to the fuzzy set A .

TYPICAL EXAMPLE: TEMPERATURE CONTROL AS A FUZZY SYSTEM

Consider the fuzzy variable TEMPERATURE, which can be described by many different adjectives each with its own fuzzy set. A typical partition of the universe of discourse, 0-40°C, is shown in Figure [3.1] where the fuzzy sets COLD, WARM and HOT are defined. Here the crisp temperature 20°C has a grade of membership of 0.5 for both COLD and WARM fuzzy sets i.e. $\mu_{cold}(20^\circ\text{C}) = \mu_{warm}(20^\circ\text{C}) = 0.5$. It is clear that the definition of fuzzy sets is non-unique and is very context-dependent. This is the nature of language and emphasizes how the actual definition of the sets are both application and user specific.

3.2.1 Classes of Membership Functions

A fuzzy set is completely characterized by its membership function (MF). Typical example of such MFs are the TRIANGULAR MF, TRAPEZOIDAL MF, GAUSSIAN MF and the GENERALIZED BELL MF, as shown in Figure [3.2]. The mathematical definitions of such membership functions and their derivatives are important for fine-tuning a fuzzy inference system to achieve a desired input/output mapping.

TRIANGULAR MFS

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

$$\text{triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (3.2)$$

by using *min* and *max*, there is an alternative expression for the preceding equation:

$$\text{triangle}(x; a, b, c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (3.3)$$

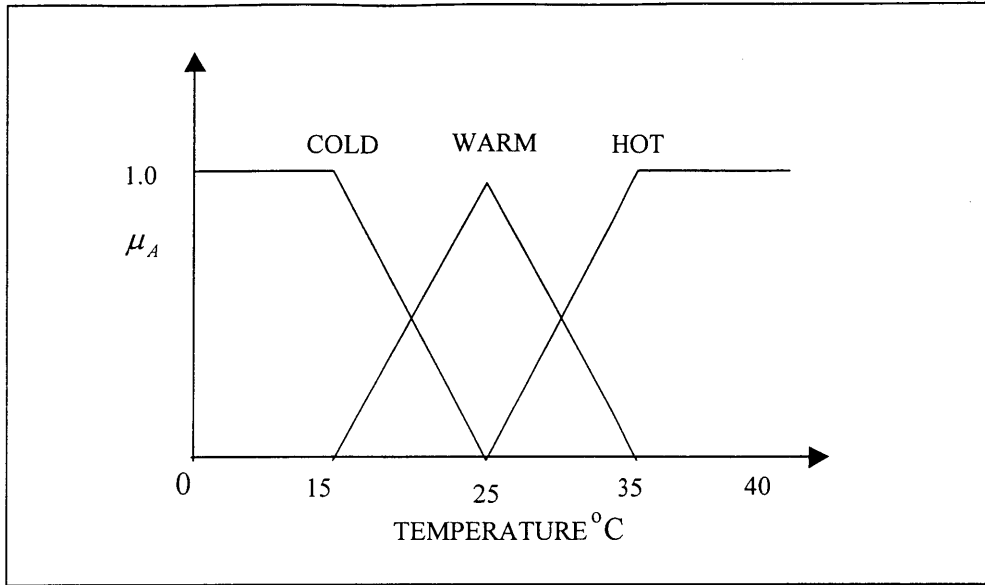


Figure [3.1] Membership functions for TEMPERATURE.

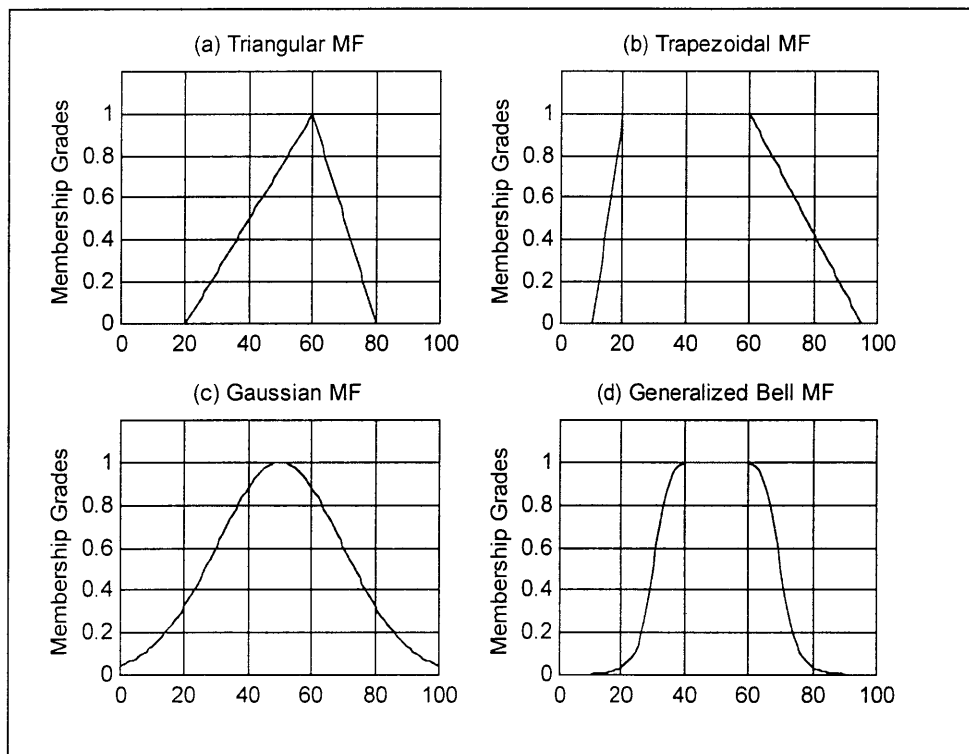


Figure [3.2] Examples of four classes of parameterized MFs: (a) Triangle($x;20,60,80$); (b) Trapeziod($x;10,20,60,95$); (c) Gaussian($x;50,20$); (d) Bell($x;20,4,50$).

The parameters $\{a, b, c\}$ (with $a < b < c$) determine the x coordinates of the three corners of the underlying triangular MF.

TRAPEZOIDAL MFS

A trapezoidal MF is specified by four parameters $\{a, b, c, d\}$ as follows:

$$\text{trapezoidal}(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (3.4)$$

an alternative concise expression using *min* and *max* is,

$$\text{trapezoidal}(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (3.5)$$

the parameters $\{a, b, c, d\}$ (with $a < b \leq c < d$) determine the x coordinates of the four corners of the underlying trapezoidal MF.

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

GAUSSIAN MFS

A Gaussian MF is specified by only two parameters $\{c, \sigma\}$:

$$\text{Gaussian}(x; c, \tau) = e^{-\frac{1}{2}\left(\frac{x-c}{\tau}\right)^2} \quad (3.6)$$

The Gaussian MF is determined completely by c and τ ; c represents the MFs center and τ determines the MFs width.

GENERALIZED BELL MFS

A generalized bell MF is specified by three parameters $\{a, b, c\}$:

$$bell(x; a, b, c) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}}, \quad (3.7)$$

where the parameter b is usually positive (If b is negative, the shape of this MF becomes an upside-down bell). Because of their smoothness and concise notation, Gaussian and bell MFs are becoming increasingly popular for specifying fuzzy sets. But they are unable to specify asymmetric MFs, which are important in certain applications. Refer to Figure [3.2] for examples of the four types of parameterized MFs.

3.3 Rule-Based Fuzzy Models

In rule-based fuzzy systems, the relationships between variables are represented by means of fuzzy IF-THEN rules of the following general form,

IF antecedent proposition **THEN** consequent proposition

The antecedent proposition is always a fuzzy proposition of the type “ x is A ” where x is a linguistic variable and A is a linguistic constant (term). The proposition’s truth value (a real number between zero and one) depends on the degree of match (similarity) between x and A . Depending on the form of the consequent two main types of rule-based fuzzy models are distinguished,

- *Linguistic fuzzy model*: both the antecedent and the consequent are fuzzy propositions.
- *Takagi-Sugeno (T-S) fuzzy model*: the antecedent is a fuzzy proposition, the consequent is a *crisp* function.

These two types of fuzzy models are detailed in the subsequent sections.

3.3.1 Linguistic fuzzy model

The linguistic fuzzy model (Zadeh [34]; Mamdani [6]) has been introduced as a way to capture available (semi-) qualitative knowledge in the form of IF-THEN rules:

$$R_i: \text{IF } x \text{ is } A_i \text{ THEN } y \text{ is } B_i \quad i = 1, 2, \dots, r \quad (3.8)$$

Here x is the input (antecedent) linguistic variable, and A_i are the antecedent linguistic terms (constants). Similarly, y is the output (consequent) linguistic variable and B_i are the consequent linguistic terms. Values of x (or y) and the linguistic terms A_i (or B_i) are fuzzy sets defined in the domains of their respective base variables: $x \in U \subset \mathfrak{R}^{F_{inp}}$ and $y \in V \subset \mathfrak{R}^{F_{out}}$.

The membership functions of the antecedent (consequent) fuzzy sets are then the mappings $\mu_A(x): U \longrightarrow [0,1]$, and $\mu_B(y): V \longrightarrow [0,1]$. Fuzzy sets A_i define fuzzy regions in the antecedent space, for which the respective consequent propositions hold. The linguistic terms A_i and B_i are usually selected from sets of predefined fuzzy terms. Denoting these sets by $\mathbf{A}$ and $\mathbf{B}$ respectively, in such away $A_i \in \mathbf{A}$ and $B_i \in \mathbf{B}$. The rule base $R = \{R_i \mid i = 1, 2, \dots, r\}$ (where r represents the total number of rules in the rule-based model), and the sets $\mathbf{A}$ and $\mathbf{B}$ constitute the knowledge base of the linguistic model.

BASIC STRUCTURE OF A FUZZY SYSTEM

A typical architecture of a fuzzy system is shown in Figure [3.3]. It is comprised of four principal components: a fuzzifier, a fuzzy rule base, an inference engine, and a defuzzifier. Details about the operations of these components are described in the following sections:

[1] Fuzzifier

A fuzzifier performs the function of a fuzzification, which can be defined as a *mapping* from an observed input space to labels of fuzzy sets in a specified input universe of discourse. Since the data manipulation is based on fuzzy set theory, fuzzification is necessary and desirable at an early stage. In fuzzy control applications, the observed data are usually crisp (through they may be corrupted by noise). A natural and simple fuzzification approach is to convert a crisp value x_0 into a fuzzy singleton A within the specified universe of discourse. That is, the membership function of A , $\mu_A(x)$, is equal to 1 at the point x_0 , and zero at other places. In this case, for a specific value $x_i(t)$ at time t , it is mapped to the fuzzy set M_{i1} with degree $\mu_1(x_i(t))$ and to the fuzzy set M_{i2} with degree $\mu_2(x_i(t))$, and so on.

There are two categories of fuzzifiers in use. They are 1) singleton, when using it, the calculation of the system's output is simplified and 2) nonsingleton, which can suppress noise in the input. A singleton fuzzifier has one point (value) x_p as its fuzzy set support, i.e., the membership function is governed by the following relation:

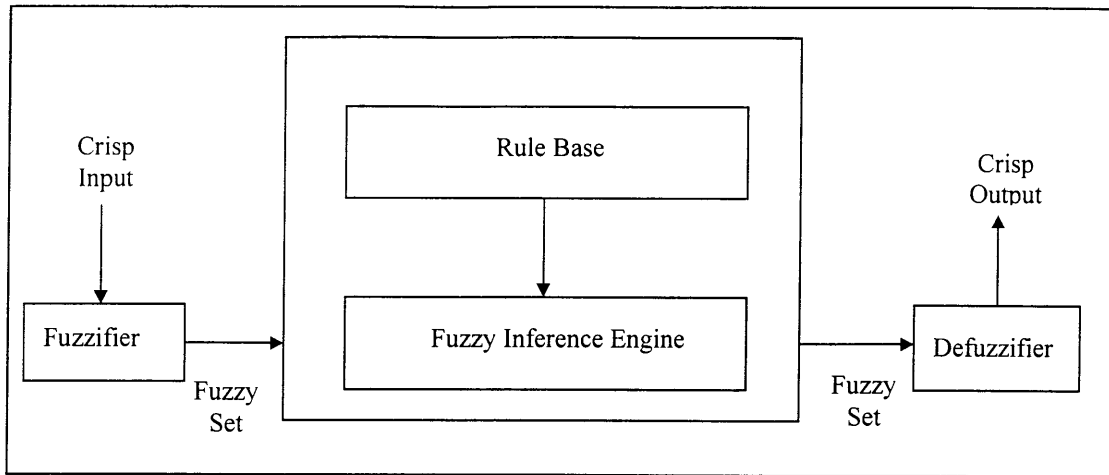


Figure [3.3] Basic architecture of a fuzzy system.

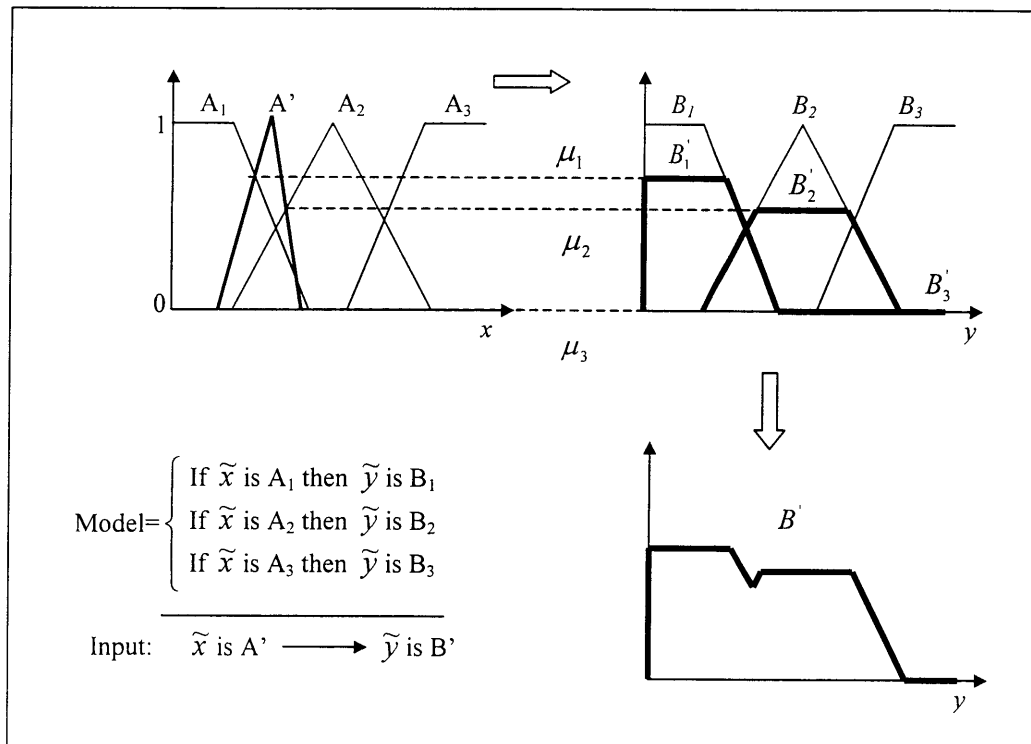


Figure [3.4] A schematic representation of the Mamdani FIS.

$$\mu_A(x) = \begin{cases} 1, & x = x_p \in U \\ 0, & x \neq x_p \in U \end{cases} \quad (3.9)$$

[2] Fuzzy Rule Base

A fuzzy rule base consists of a set of fuzzy IF-THEN rules. It is the heart of the fuzzy system in the sense that all other components are used to implement these rules in a reasonable and efficient manner. The general form of the fuzzy control rules in the case of multi-input-single-output systems (MISO) is:

$$\text{IF } x \text{ is } A_i, \dots \text{ AND } y \text{ is } B_i, \text{ THEN } z \text{ is } C_i \quad i=1,2,\dots,r \quad (3.10)$$

In which $x, \dots, y$ and z are linguistic variables representing the system state variables and control variables, respectively, and $A_i, \dots, B_i$ and C_i are the linguistic values of the linguistic variables $x, \dots, y$ and z in the universes of discourse $U, \dots, V$, and W , respectively, “AND,” “OR,” and “NOT” are connectives of the rule.

In general there are many forms and variations of fuzzy rules. For instance, the following form whose consequent is represented as a function of a system state variables $x, \dots, y$,

$$R_i: \text{IF } x \text{ is } A_i, \dots \text{ AND } y \text{ is } B_i, \text{ THEN } z = f(x, \dots, y) \quad i=1,2,\dots,r \quad (3.11)$$

Fuzzy control rules evaluate a system state (i.e. state, state error, state error integral, and so on) at time t and compute and decide control actions as a function of the state variables $(x, \dots, y)$. This type of rules does not need a defuzzifier to decode any linguistic fuzzy values into numerical values. Such fuzzy model is called Takagi-Sugeno model will be explained later.

[3] Fuzzy Inference System (FIS)

In order to be able to employ a linguistic model, we need an algorithm which allows us to compute an output value given some input value. This algorithm is called the fuzzy inference algorithm (or mechanism or engine) such as Mamdani FIS shown in Figure [3.4] with three rules. For the linguistic model, the inference mechanism can be derived using fuzzy relational calculus.

The fuzzy inference engine performs the decision-making aspect of the system. A complete rule base with fuzzy input sets F_{inp} and fuzzy output sets F_{out} consists of $F_{inp} \times F_{out}$ rules each

with its own membership function $\mu_{r_{ij}}(\mathbf{x}, y)$ defined on the product space $\{A \times B\} \subset R^{n+1}$. The resulting rule base $\mu_R(\mathbf{x}, y)$ is hence defined as the union of these membership functions:

$$\mu_R(\mathbf{x}, y) = \bigcup_{ij} IF \mu_{A_i}(\mathbf{x}) \text{ THEN } \mu_{B_j}(y) \text{ c}_{ij} \quad (3.12)$$

$$\begin{aligned} &= \bigcup_{ij} \mu_{r_{ij}}(\mathbf{x}, y) \\ &= s(t(\mu_{A_1}(\mathbf{x}), \mu_{B_1}(y), c_{11}), \dots, t(\mu_{A_{F_{inp}}}(\mathbf{x}), \mu_{B_{F_{out}}}(y), c_{F_{inp}F_{out}}))), \end{aligned} \quad (3.13)$$

which is also known as the fuzzy relational surface. From this rule base a fuzzy output is inferred by the computational rule of inference defined as:

$$\begin{aligned} \mu_B(y) &= \mu_{A'}(x) \circ \mu_R(x, y) \\ &= \int_X \mu_{A'}(x) \circ \mu_R(x, y) dx \end{aligned} \quad (3.14)$$

where $\circ$ is the composition operator, $\mu_{A'}(x)$ is the fuzzified input and $\mu_B(y)$ represents the distribution of the fuzzy membership functions across the output. Using singleton fuzzification this reduces to a slice through the fuzzy relational surface. The result is a fuzzy output representing the degree to which the input is related to each of the fuzzy output sets.

[4] Defuzzifier

Defuzzification is a mapping from a space of fuzzy control actions defined over an output universe of discourse into a space of nonfuzzy (crisp) control actions. There are many different defuzzification techniques, the most natural is the center of gravity or center of area where the crisp model's output is found from,

$$y_c = \frac{\sum_{i=1}^r y_i \mu_{A_i}(y_i)}{\sum_{i=1}^r \mu_{A_i}(y_i)} \quad (3.15)$$

MULTIVARIABLE SYSTEMS

In the MIMO case, all fuzzy sets in the model are defined on vector domains by multivariate membership functions. It is, however, usually, more convenient to write the antecedent and consequent propositions as logical combinations of fuzzy propositions with univariate membership functions. Fuzzy logic operators, such as the conjunction, disjunction and negation (complement), can be used to combine the propositions. Furthermore, a MIMO

model can be written as a set of MISO models. Therefore, for the ease of notation, we will write the rules for MISO systems. Most common is the *conjunctive form* of the antecedent, represented by:

$$\begin{aligned} R_i : \text{IF } x_1 \text{ is } A_{i1} \text{ AND } x_2 \text{ is } A_{i2} \text{ AND } \dots \text{ AND } x_p \text{ is } A_{ip} \\ \text{THEN } y \text{ is } B_i, \quad i=1,2,\dots,r \end{aligned} \quad (3.16)$$

The above model is a special case of Equ. [3.8], as the fuzzy set A_i in Equ. [3.8] is obtained as the Cartesian product of fuzzy sets A_{ij} : $A_i = A_{i1} \times A_{i2} \times \dots \times A_{iF_{inp}}$. The degree of fulfillment μ_i is given by:

$$\mu_i = \mu_{A_{i1}}(x_1) \wedge \mu_{A_{i2}}(x_2) \wedge \dots \wedge \mu_{A_{iF_{inp}}}(x_{F_{inp}}) \quad 1 \leq i \leq r \quad (3.17)$$

Other conjunction operators, such as the product, can be used. A set of rules in the conjunctive antecedent form divides the input domain into a lattice of fuzzy hyperboxes, parallel with the axes. Each of the hyperboxes is a Cartesian product-space intersection of the corresponding univariate fuzzy sets. This is shown in Figure [3.5a]. The number of rules in the conjunctive form needed to cover the entire domain is given by,

$$r = \prod_{i=1}^{F_{inp}} N_i \quad (3.18)$$

where F_{inp} is the dimension of the input space and N_i is the number of linguistic terms of the i^{th} antecedent variable.

By combining conjunctions, disjunctions and negations, various partitions of the antecedent space can be obtained, the boundaries are, however, restricted to the rectangular grid defined by the fuzzy sets of the individual variables, see Figure [3.5b]. For instance, considering the rule antecedent covering the lower left corner of the antecedent space in the figure,

$$R_i : \text{IF } x_1 \text{ is NOT } A_{i3} \text{ AND } x_2 \text{ is } A_{i2} \text{ THEN } \dots$$

Degree of fulfillment of such a rule is computed using the complement and intersection operators,

$$\mu = [1 - \mu_{A_{i3}}(x_1)] \wedge \mu_{A_{i2}}(x_2) \quad (3.19)$$

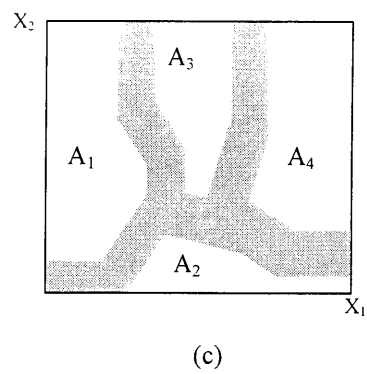
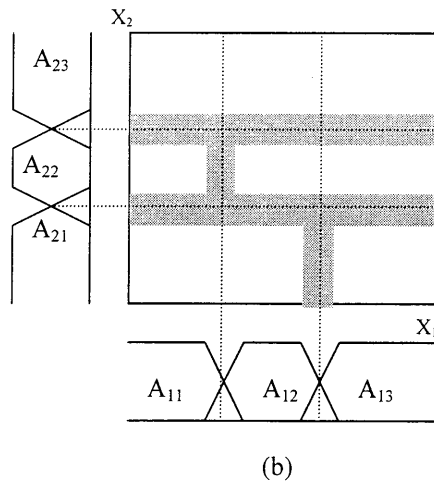
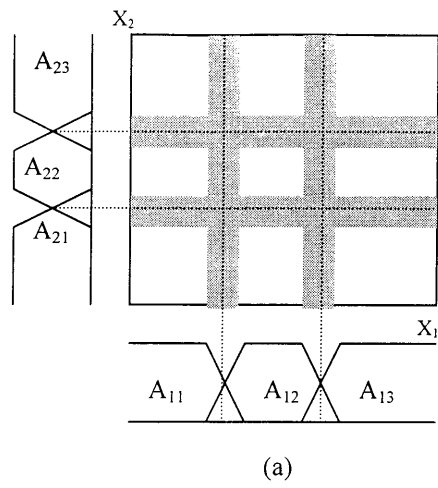


Figure [3.5] Different partitions of the antecedent space. Gray areas denote the overlapping regions of the fuzzy sets.

The antecedent form with multivariate membership functions Equ. [3.8] is the most general one, as there is no restriction on the shape of the fuzzy regions. Boundaries between these regions can be arbitrarily curved and opaque to the axes, as depicted in Figure [3.5c]. Also the number of fuzzy sets needed to cover the antecedent space may be much smaller than in the previous cases. Hence, for complex multivariable systems, this partition may provide the most effective representation. Note that the fuzzy sets A_1 to A_4 in Figure [3.5c] still can be projected onto x_1 and x_2 to obtain an approximate linguistic interpretation of the regions described.

3.3.2 Takagi–Sugeno Fuzzy Model

The linguistic model, introduced in the previous section, describes a given system by means of linguistic IF-THEN rules with fuzzy proposition in the antecedent as well as in the consequent. Takagi–Sugeno (T-S) fuzzy model [49], on the other hand, uses crisp functions in the consequents. Hence, it can be seen as a combination of linguistic and mathematical regression modeling in the sense that the antecedents describe fuzzy regions in the input space in which consequent functions are valid. A typical T-S rules have the following form:

$$R_i: \text{IF } \mathbf{x} \text{ is } A_i \text{ THEN } \mathbf{y}_i = f_i(\mathbf{x}) \quad (3.20)$$

Contrary to the linguistic model, the input $\mathbf{x}$ is a crisp variable (linguistic inputs are in principle possible, but would require the use of the extension principle [56] to compute the fuzzy value of $\mathbf{y}_i$). The functions f_i are typically of the same structure, only the parameters in each rule are different. Generally, f_i is a vector-valued function, but for the ease of notation we will consider a scalar f_i . A simple and practically useful parameterization is the affine (linear in parameters) form, yielding the rules,

$$\text{IF } \mathbf{x} \text{ is } A_i \text{ THEN } y_i = \mathbf{a}_i^T \mathbf{x} + b_i \quad i=1,2,\dots,r \quad (3.21)$$

In Equ. [3.21] $\mathbf{a}_i$ is a parameter vector and b_i is a scalar offset. This model is called an affine T-S model.

INFERENCE MECHANISM OF T-S FUZZY MODEL

The inference formula of the T-S fuzzy model is a straightforward extension of the singleton model inference,

$$\begin{aligned}
y &= \frac{\sum_{i=1}^r \mu_i y_i}{\sum_{i=1}^r \mu_i} \\
&= \frac{\sum_{i=1}^r \mu_i (\mathbf{a}_i^T \mathbf{x} + b_i)}{\sum_{i=1}^r \mu_i}
\end{aligned} \tag{3.22}$$

once the antecedent fuzzy sets define distinct but overlapping regions in the antecedent space and the parameters $\mathbf{a}_i$ and b_i correspond to a *local linearization* of a nonlinear function, the T-S model can be regarded as a smoothed piece-wise approximation of that function. Figure [3.6] shows a nonlinear function that can be described by three different affine equations.

T-S MODEL AS A QUASI-LINEAR SYSTEMS

The affine T-S model can be regarded as a quasi-linear system (i.e., a linear system with input-dependent parameters). Denoting the normalized degree of fulfillment by

$$h_i(\mathbf{x}) = \mu_i(\mathbf{x}) / \sum_{j=1}^r \mu_j(\mathbf{x}) \tag{3.23}$$

Here $h_i(\mathbf{x})$ can be written explicitly as a function of $\mathbf{x}$ to show that the T-S fuzzy model is a quasi-linear model of the following form:

$$\begin{aligned}
y &= \left(\sum_{i=1}^r h_i(\mathbf{x}) \mathbf{a}_i^T \right) \mathbf{x} + \sum_{i=1}^r h_i(\mathbf{x}) b_i \\
&= \mathbf{a}^T(\mathbf{x}) \mathbf{x} + b(\mathbf{x}).
\end{aligned} \tag{3.24}$$

where the ‘parameters’ $\mathbf{a}(\mathbf{x})$, $b(\mathbf{x})$ are convex linear combinations of the consequent parameters $\mathbf{a}_i$ and b_i , defined by,

$$\mathbf{a}(\mathbf{x}) = \sum_{i=1}^r h_i(\mathbf{x}) \mathbf{a}_i, \quad b(\mathbf{x}) = \sum_{i=1}^r h_i(\mathbf{x}) b_i \tag{3.25}$$

In this sense, a T-S model can be regarded as a mapping from the antecedent (input) space to a convex region (polytope) in the space of the parameters of a quasi-linear system, as schematically depicted in Figure [3.7].

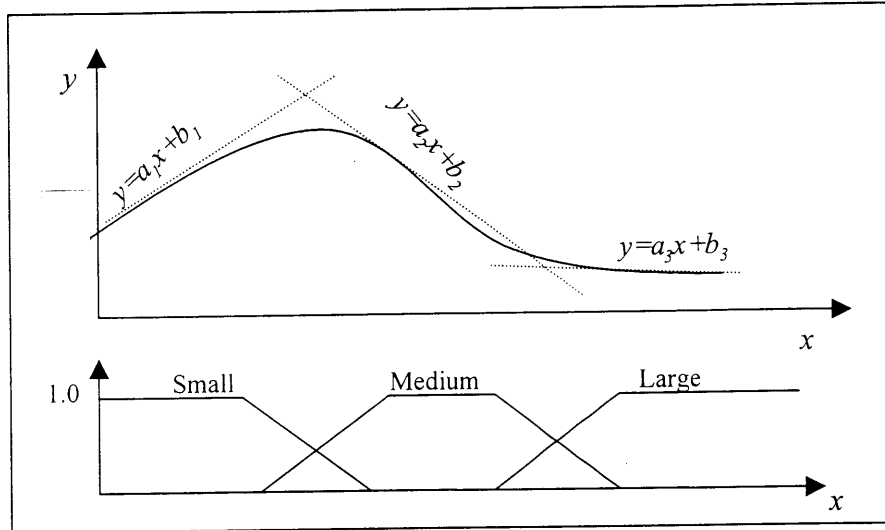


Figure [3.6] T-S fuzzy model as a smoothed piece-wise linear approximation of a nonlinear function.

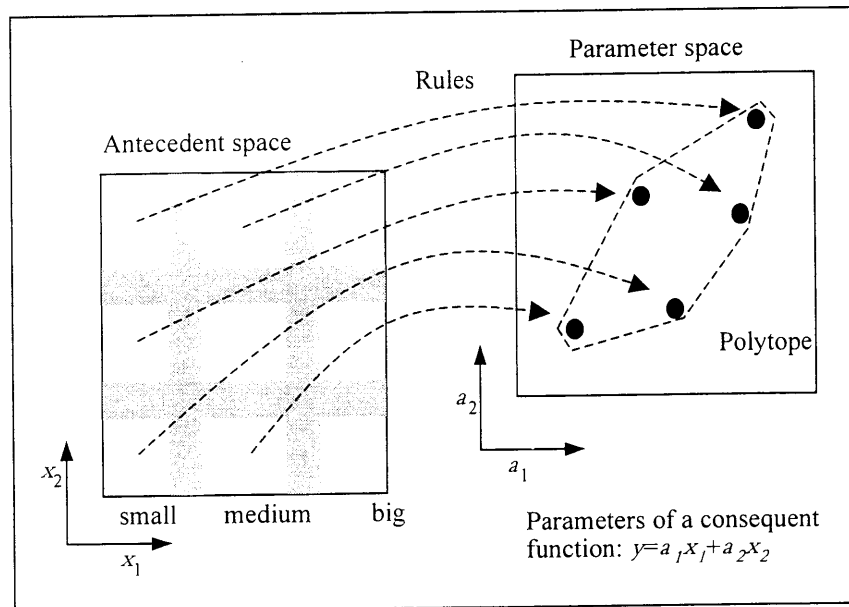


Figure [3.7] A T-S model with affine consequent can be regarded as a mapping from the antecedent space to the consequent parameters.

This property facilitates the analysis of T-S models in a framework similar to that of linear systems. Methods have been developed to design controllers with desired closed loop characteristics and also to analyze their stability, as were introduced in chapter one.

3.4 Artificial Neural Networks

Having discussed the background of fuzzy logic and systems, it is essential to highlight as well the essentials of neural networks as they constitute the fundamentals of multilayer perceptron.

3.4.1 Multilayer perceptron

The best known neural network architecture is the multilayer feedforward neural network (*multilayer preceptron*). It is a static network that consists of a number of layers: input layer, output layer and one or more hidden layers connected in a feedforward way. Each layer consists of a number of neurons. One single neuron makes the simple operation of a weighted sum of the incoming signals and a bias term (or *threshold*), fed through an activation function and resulting in the output value of the neuron.

A network with one hidden layer is described in matrix-vector notation as,

$$\mathbf{y} = W a(V\mathbf{x} + \beta) \quad (3.26)$$

or in the elementwise notation:

$$y_i = \sum_{r=1}^{n_h} w_{ir} a\left(\sum_{j=1}^m v_{rj} x_j + \beta_r\right), \quad i=1, \dots, n_o \quad (3.27)$$

Here $\mathbf{x} \in \mathbb{R}^m$ is the input and $\mathbf{y} \in \mathbb{R}^{n_o}$ the output of the network and the nonlinear operation $a(\cdot)$ is taken elementwise. The interconnections matrices are $W \in \mathbb{R}^{n_o \times n_h}$ for the output layer, $V \in \mathbb{R}^{n_h \times m}$ for the hidden layer, $\beta \in \mathbb{R}^{n_h}$ is the bias vector (thresholds of hidden neurons) with n_h the number of hidden neurons.

For a network with two hidden layers (see Figure [3.8]) $\mathbf{y}$ is calculated as,

$$\mathbf{y} = W a(V_2 a(V_1 \mathbf{x} + \beta_1) + \beta_2), \quad (3.28)$$

or

$$y_i = \sum_{r=1}^{n_{h2}} w_{ir} a\left(\sum_{s=1}^{n_{h1}} v_{rs}^{(2)} a\left(\sum_{j=1}^m v_{sj}^{(1)} x_j + \beta_s^{(1)}\right) + \beta_r^{(2)}\right), \quad i=1, \dots, n_o \quad (3.29)$$

The interconnection matrices are $\mathbf{W} \in \mathbb{R}^{n_o \times n_{h_2}}$ for the output layer, $\mathbf{V}_2 \in \mathbb{R}^{n_{h_2} \times n_{h_1}}$ for the second hidden layer and $\mathbf{V}_1 \in \mathbb{R}^{n_{h_1} \times m}$ for the first hidden layer. The bias vectors are $\beta_2 \in \mathbb{R}^{n_{h_2}}, \beta_1 \in \mathbb{R}^{n_{h_1}}$ for the second and first hidden layer respectively. In order to describe a network with N_{layer} layers ($N_{\text{layer}} - 1$ hidden layers, because the input layer is a ‘dummy’ layer), the following notation will be used in the sequel,

$$x_i^l = a(\xi_i^l), \quad \xi_i^l = \sum_{j=1}^{N_l} w_{ij}^l x_j^{l-1} \quad (3.30)$$

where $l=1, \dots, N_{\text{layer}}$ is the layer index, N_l denotes the number of neurons in layer l and x_i^l is the output of the neurons at layer l . The thresholds are considered here to be part of the interconnection matrix, by defining additional constant inputs.

The choice of activation function $a(\cdot)$ may depend on the application area. Typical activation functions are shown in Figure [3.9]. For applications in modeling and control the hyperbolic tangent function,

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (3.31)$$

is normally used. In case of a ‘ $\tanh$ ’ the derivative of the activation function $a' = 1 - a^2$ is used. The neurons of the input layer have a linear activation function.

A class of neural network that has received a lot of attention recently in the field of neural networks is the Radial Basis Function Network RBFN. This network can be described as,

$$\mathbf{y} = \sum_{i=1}^{n_h} w_i g(\|\mathbf{x} - \mathbf{c}_i\|), \quad (3.32)$$

with $\mathbf{x} \in \mathbb{R}^m$ the input vector and $\mathbf{y} \in \mathbb{R}$ the output (models with multiple outputs are also possible). The network consists of one hidden layer n_h hidden neurons. One of the basic differences with the multilayer perceptron is in the use of the activation function: in many cases one takes a Gaussian function for $g(\cdot)$, which is radially symmetric with respect to the input argument. The output layer has output weights $w \in \mathbb{R}^{n_h}$. The parameters for the hidden layer are the centers $\mathbf{c}_i \in \mathbb{R}^m$.

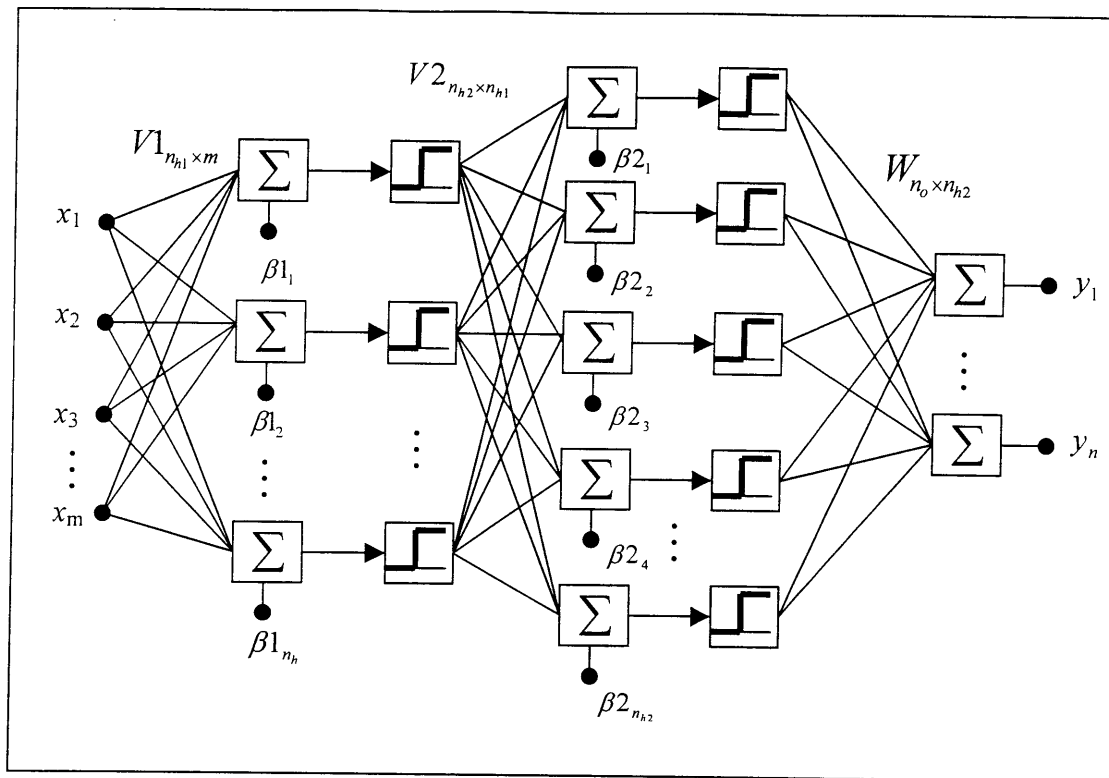


Figure [3.8] An Artificial Neural Network with two hidden layers.

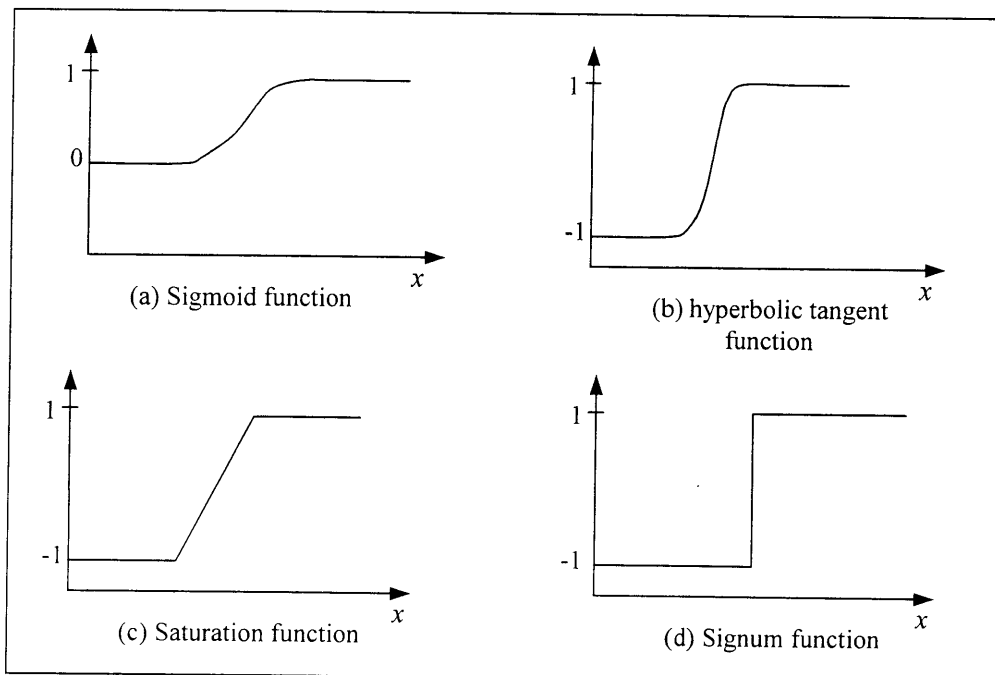


Figure [3.9] Some possible activation functions for the neurons in the multilayer perceptron.

The *multilayer perceptron* and the *radial basis* networks are feedforward neural networks, which means that there is a static nonlinear mapping from the input space to the output space, the layers are interconnected in a feedforward way to each other.

3.4.2 Back-propagation Learning Algorithm

The back-propagation learning algorithm is one of the most important historical developments in neural networks. This learning algorithm is applied to multilayer feedforward networks consisting of processing elements with continuous differentiable activation functions. Such networks associated with the back-propagation learning algorithm are also called back-propagation networks.

For a given training set of input-output pairs $\{(\mathbf{x}^{(k)}, \mathbf{d}^{(k)})\}$, $k = 1, 2, \dots, n_{\text{data}}$, the algorithm provides a procedure for changing the weights in a back-propagation network to classify the given input patterns correctly. The basis for this weight update algorithm is simply the *Gradient-Decent* method as used for simple perceptron with differentiable units.

For a given input-output pair $\{(\mathbf{x}^{(k)}, \mathbf{d}^{(k)})\}$, the back-propagation algorithm performs two phases of data flow. First, the input pattern $\mathbf{x}^{(k)}$ is propagated from the input layer to the output layer, and as a result of this forward flow of data, it produces an actual output $\mathbf{y}^{(k)}$. Hence, the error signals resulting from the difference between $\mathbf{d}^{(k)}$ and $\mathbf{y}^{(k)}$ are back propagated from the output layer to the previous layers for them to update their weights. To illustrate the details of the back-propagation learning algorithm, consider a three-layer network as shown in Figure [3.10].

The results can be easily extended to networks with any number of layers. In Figure [3.10], we have m PEs (Processing Element) in the input layer, l PEs in the hidden layer, and n PEs in the output layer; the solid lines show the forward propagation of signals, and the dashed lines show the backward propagation of errors.

To illustrate the mechanism of learning, consider an input-output training pair (x, d) , where the superscript k is omitted for notation simplification.

Given an input pattern x , a PE q in the hidden layer receives a net input of,

$$net_q = \sum_{j=1}^m v_{qj} x_j \quad (3.34)$$

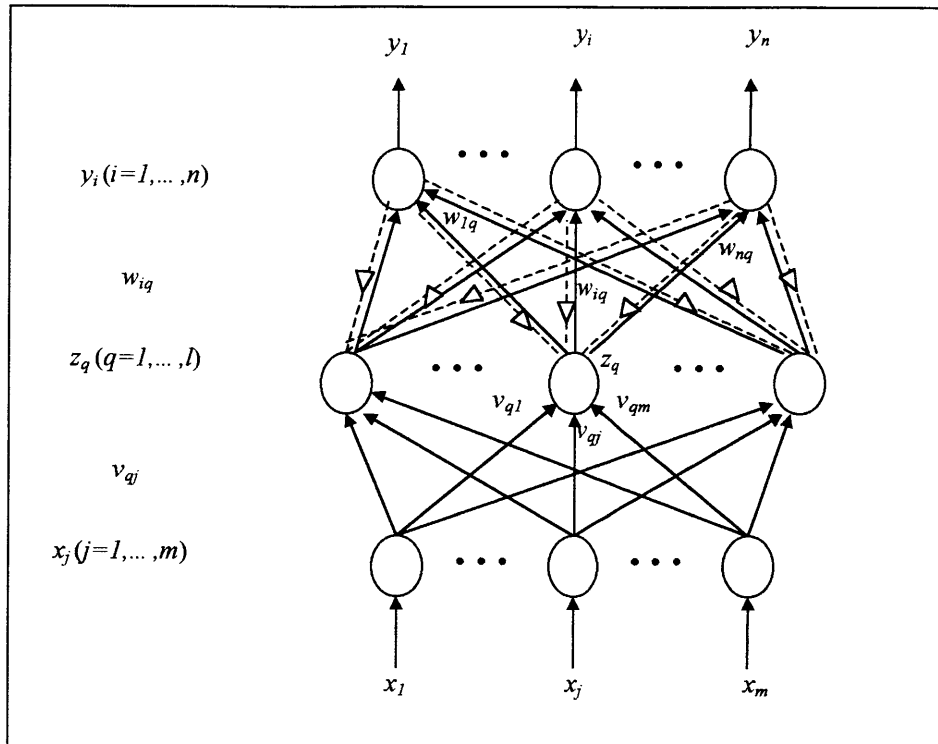


Figure [3.10] Three-layer back-propagation network

and produces an output of

$$z_q = a(\text{net}_q) = a\left(\sum_{j=1}^m v_{qj} x_j\right) \quad (3.35)$$

The net input for a processing element i in the output layer is then

$$\begin{aligned} \text{net}_i &= \sum_{q=1}^l w_{iq} z_q \\ &= \sum_{q=1}^l w_{iq} a\left(\sum_{j=1}^m v_{qj} x_j\right), \end{aligned} \quad (3.36)$$

and it produces an output of

$$\begin{aligned} y_i &= a(\text{net}_i) = a\left(\sum_{q=1}^l w_{iq} z_q\right) \\ &= a\left(\sum_{q=1}^l w_{iq} a\left(\sum_{j=1}^m v_{qj} x_j\right)\right) \end{aligned} \quad (3.37)$$

The above equations indicate the forward propagation of input signals through the layers of neurons. Hence, we need to consider the error signals and their back propagation. Defining a cost function $E(\mathbf{w})$ given as,

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2} \sum_{i=1}^n (d_i - y_i)^2 \\ &= \frac{1}{2} \sum_{i=1}^n (d_i - a(\text{net}_i))^2 \end{aligned} \quad (3.38)$$

$$= \frac{1}{2} \sum_{i=1}^n \left[d_i - a\left(\sum_{q=1}^l w_{iq} z_q\right) \right]^2. \quad (3.39)$$

In accordance to the gradient-descent method, weights in the hidden-to-output connections are updated in accordance to,

$$\Delta w_{iq} = -\eta \frac{\partial E}{\partial w_{iq}} \quad (3.40)$$

Using Eqs. ([3.36]-[3.38]) and the chain rule for $\frac{\partial E}{\partial w_{iq}}$, we have

$$\Delta w_{iq} = -\eta \left[\frac{\partial E}{\partial y_i} \right] \left[\frac{\partial y_i}{\partial \text{net}_i} \right] \left[\frac{\partial \text{net}_i}{\partial w_{iq}} \right]$$

$$\begin{aligned}
&= \eta [d_i - y_i] [a'(net_i)] [z_q] \\
&\triangleq \eta \delta_{oi} z_q,
\end{aligned} \tag{3.41}$$

where δ_{oi} is the error signal and its double subscript indicates the i^{th} node in the output layer.

The error signal is defined by,

$$\begin{aligned}
\delta_{oi} &\triangleq - \frac{\partial E}{\partial net_i} \\
&= - \left[\frac{\partial E}{\partial y_i} \right] \left[\frac{\partial y_i}{\partial net_i} \right] \\
&= [d_i - y_i] [a'(net_i)]
\end{aligned} \tag{3.42}$$

where net_i is the net input to PE i of the output layer and $a'(net_i) = \frac{\partial a(net_i)}{\partial net_i}$.

The result is identical to the delta learning rule for a single-layer perceptron whose input is now the output z_q of the hidden layer.

For the weight update on the input-to-hidden connections, we use the chain rule with the gradient-descent method and obtain the weight update on the link weight connecting PE j in the input layer to PE q in the hidden layer,

$$\begin{aligned}
\Delta v_{qj} &= -\eta \left[\frac{\partial E}{\partial v_{qj}} \right] \\
&= -\eta \left[\frac{\partial E}{\partial net_q} \right] \left[\frac{\partial net_q}{\partial v_{qj}} \right] \\
&= -\eta \left[\frac{\partial E}{\partial z_q} \right] \left[\frac{\partial z_q}{\partial net_q} \right] \left[\frac{\partial net_q}{\partial v_{qj}} \right]
\end{aligned} \tag{3.43}$$

From Equ. [3.39], each error term $[d_i - y_i]$, $i=1,2,\dots,n$, is a function of z_q .

Evaluating the chain rule, we have

$$\Delta v_{qj} = \eta \sum_{i=1}^n [(d_i - y_i) a'(net_i) w_{iq}] a'(net_q) x_j \quad (3.44)$$

Using Equ. [3.42], we can write Equ. [3.44] as

$$\begin{aligned} \Delta v_{qj} &= \eta \sum_{i=1}^n [(d_{oi} w_{iq}) a'(net_q) x_j] \\ &= \eta \delta_{hq} x_j, \end{aligned} \quad (3.45)$$

where δ_{hq} is the error signal of PE q in the hidden layer and is defined as

$$\begin{aligned} \delta_{hq} &\triangleq - \frac{\partial E}{\partial net_q} \\ &= - \left[\frac{\partial E}{\partial z_q} \right] \left[\frac{\partial z_q}{\partial net_q} \right] \\ &= a'(net_q) \sum_{i=1}^n \delta_{oi} w_{iq}, \end{aligned} \quad (3.46)$$

where net_q is the net input to the hidden PE q Equ. [3.34]. The error signal of a PE in a hidden layer is different from the error signal of a PE in the output layer, as seen in Eqs. [3.42] and [3.46].

Because of this difference, the above weight update procedure is called the generalized delta rule. It is observed from Equ. [3.46] that the error signal δ_{hq} of a hidden PE q can be determined in terms of the error signal δ_{oi} of the PEs output y_i , that it feeds.

The coefficients are just the weights used for the forward propagation, but here they are propagating error signals (δ_{oi}) backward instead of propagating signals forward. This is shown by the dashed lines in Figure [3.10]. In addition, it demonstrates one important feature of the back-propagation algorithm, the update rule is local; that is, to compute the weight change for a given connection, we need only quantities available at both ends of that connection. The above derivation can be easily extended to the network with more than one hidden layer by using the chain rule continuously. In general, with an arbitrary number of layers, the back-propagation update rule is in the form:

$$\begin{aligned}\Delta w_{ij} &= \eta \delta_i x_j, \\ &= \eta \delta_{\text{output}-i} \cdot x_{\text{input}-j}\end{aligned}\quad (3.47)$$

where “output- i ” and “input- j ” refer to the two ends of the connection from PE j to PE i , x_j is the proper input-end activation from a hidden PE or an external input, and δ_i is the learning signal which is defined by Equ. [3.42] for the last (or output) layer of connection weights and defined by Equ. [3.46] for all the other layers. In summary, the back-propagation learning can be outlined in the following algorithm.

BACK-PROPAGATION LEARNING RULE ALGORITHM

Considering a neural network with Q feedforward layers, $q=1,2,\dots,Q$ and let ${}^q\text{net}_i$ and ${}^q y_i$ denote the net input and output of the i^{th} unit in the q^{th} layer, respectively. The network has m input nodes and n output nodes. Let ${}^q w_{ij}$ denote the connection weight from ${}^{q-1} y_j$ to ${}^q y_i$.

INPUT: A set of training pairs $\{(\mathbf{x}^{(k)}, \mathbf{d}^{(k)}) \mid k = 1, 2, \dots, n_{\text{data}}\}$, where the input vectors are augmented with the last elements as -1 , that is, $x_{m+1}^{(k)} = -1$.

STEP 0 (Initialization): Choose $\eta > 0$ and E_{max} (maximum tolerable error). Initialize the weights to small random values. Set $E = 0$ and $k = 1$.

STEP 1 (Training loop): Apply the k^{th} input pattern to the input layer ($q=1$):

$${}^q y_i = {}^1 y_i = x_i^{(k)} \quad \text{for all } i.$$

STEP 2 (Forward propagation): Propagate the signal forward through the network using

$$\begin{aligned}{}^q y_i &= a({}^q \text{net}_i) \\ &= a\left(\sum_j {}^q w_{ij} {}^{q-1} y_j\right)\end{aligned}$$

for each i and q until the outputs of the output layer ${}^Q y_i$ have all been obtained.

STEP 3 (Output error measure): Compute the error value and error signals ${}^Q \delta_i$ for the output

layer:

$$E = \frac{1}{2} \sum_{i=1}^n (d_i^{(k)} - {}^Q y_i)^2 + E,$$

$${}^Q\delta_i = (d_i^{(k)} - y_i) a'({}^Q net_i)$$

STEP 4 (Error back-propagation): Propagate the error backward to update the weights and compute the error signals ${}^{q-1}\delta_i$ for the preceding layers:

$$\begin{aligned} \Delta^q w_{ij} &= \eta^q \delta_i {}^{q-1}y_j & \text{and } {}^q w_{ij}^{new} &= {}^q w_{ij}^{old} + \Delta^q w_{ij}, \\ {}^{q-1}\delta_i &= a'({}^{q-1} net_i) \sum_j {}^q w_{ij} {}^q \delta_j & \text{for } q = Q, Q-1, \dots, 2. \end{aligned}$$

STEP 5 (one epoch looping): Check whether the whole set of training data has been cycled one. If $k < p$, then $k = k + 1$ and go to step 1; otherwise, go to step 6.

STEP 6 (Total error checking): Check whether the current total error is acceptable: If $E < E_{max}$, then terminate the training process and output the final weights; otherwise, $E=0$, $k=1$, and initiate the new training epoch by going to step 1.

The presented algorithm methodology adopts the incremental approach in updating the weights; that is, the weights are changed immediately after training pattern presented. The alternative is batch-mode training, where the weights are changed only after all training patterns have been presented. The relative effectiveness of the two approaches depends on the problem, but batch-mode training requires additional local storage for each connection to maintain the immediate weight change.

3.5 ANFIS: Adaptive Neuro-Fuzzy Inference Systems

Having talked about the fuzzy system and artificial neural networks separately, here it is the time to present their integration. In this section we will concentrate on the ANFIS architectures for the first order T-S fuzzy model because of its transparency and efficiency.

For simplicity, we assume that the fuzzy inference system under consideration has two inputs x and y and one output z . For a first-order T-S fuzzy model, a common rule set with two fuzzy IF-THEN rules is the following:

$$R_1: \text{ IF } x \text{ is } A_1 \text{ AND } y \text{ is } B_1, \text{ THEN } f_1 = p_1 x + q_1 y + r_1 \quad (3.49)$$

$$R_2: \text{ IF } x \text{ is } A_2 \text{ AND } y \text{ is } B_2, \text{ THEN } f_2 = p_2 x + q_2 y + r_2 \quad (3.50)$$

Figure [3.11a] illustrates the reasoning mechanism for this T-S model.

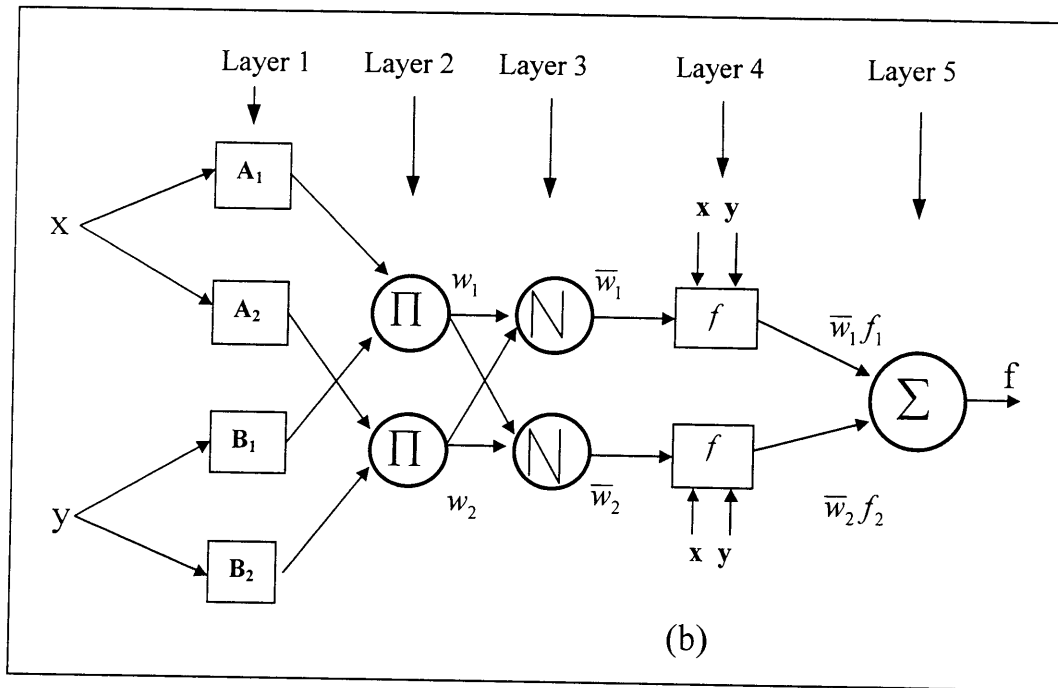
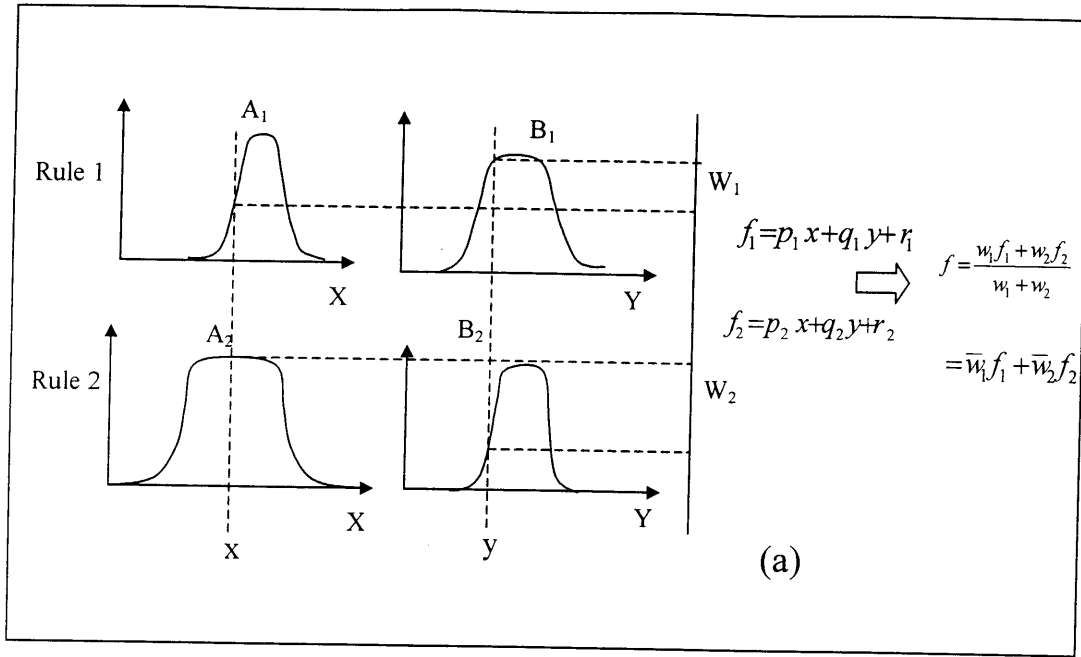


Figure [3.11] (a) A two-input first-order Takagi-Sugeno fuzzy model with two rules; (b) equivalent ANFIS architecture.

3.5.1 ANFIS Architecture

The corresponding equivalent ANFIS architecture is as shown in Figure [3.11b], where nodes of the same layer have similar functions, as described next. (Here we denote the output of the i^{th} node in layer l as $O_{l,i}$.)

LAYER 1 every node i in this layer is an adaptive node with a node function

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x), & \text{for } i=1,2, \text{ or} \\ O_{1,i} &= \mu_{B_{i-2}}(y) & \text{for } i=3,4, \end{aligned}$$

where x (or y) is the input to node i and A_i (or B_{i-2}) is a linguistic label (such as “small” or “large”) associated with this node. In other words, O_1 is the membership grade of a fuzzy set $A = (A_1, A_2, B_1 \text{ or } B_2)$ and it specifies the degree to which the given input x (or y) satisfies the quantifier A . Here the membership function for A can be any appropriate parameterized membership function, such as the generalized bell function:

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b}},$$

where $\{a_i, b_i, c_i\}$ is the parameter set. As the values of these parameters change, the bell-shaped function varies accordingly, thus exhibiting various forms of membership functions for fuzzy set A . Parameters in this layer are referred to as premise parameters.

LAYER 2 every node in this layer is a fixed node labeled $\prod$, whose output is the product of all the incoming signals:

$$\begin{aligned} Q_{2,i} &= w_i, \\ &= \mu_{A_i}(x) \mu_{B_i}(y) & i=1,2 \end{aligned}$$

Each node output represents the firing strength of a rule. In general, any other T-norm operators that perform fuzzy AND can be used as the node function in this layer.

LAYER 3 every node in this layer is a fixed node labeled N . The i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of all rules' firing strengths:

$$Q_{3,i} = \bar{w}_i$$

$$= \frac{w_i}{w_1 + w_2} \quad i=1,2$$

for convenience, outputs of this layer are called normalized firing strengths.

LAYER 4 every node i in this layer is an adaptive node with a node function

$$\begin{aligned} Q_{4,i} &= \bar{w}_i f_i \\ &= \bar{w}_i (p_i x + q_i y + r_i) \end{aligned}$$

where $\bar{w}_i$ is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

LAYER 5 the single node in this layer is a fixed node labeled Σ , which computes the overall output as the summation of all incoming signals:

$$\begin{aligned} \text{Overall output} &= Q_{5,1} = \sum_i \bar{w}_i f_i \\ &= \frac{\sum_i w_i f_i}{\sum_i w_i} \end{aligned}$$

Thus we have constructed an adaptive network that is functionally equivalent to a T-S fuzzy model. The next sub-section will presents the learning algorithm for the five layer ANFIS network just explained above.

3.5.2 ANFIS Learning Algorithm

From the ANFIS architecture shown in Figure [3.11b], we observe that as the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters. In symbols, the output f in Figure [3.11b] can be rewritten as,

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \\ &= \bar{w}_1 (p_1 x + q_1 y + r_1) + \bar{w}_2 (p_2 x + q_2 y + r_2) \\ &= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2 \end{aligned} \quad (3.51)$$

which is linear in the consequent parameters $p_1, q_1, r_1, p_2, q_2,$ and r_2 .

From this observation, we have

S = set of total parameters,

S_1 = set of premise (nonlinear parameters)

S_2 = set of consequent (linear) parameters

A hybrid algorithm adjusts the consequent parameters $S_2 = \{p_1, q_1, r_1, p_2, q_2, r_2\}$ in a forward pass and the premise parameters $S_1 = \{a_i, b_i, c_i\}$ in a backward pass (Jang et al.) [2]. In the forward pass the network inputs propagate forward until layer 4, where the consequent parameters are identified by the *Least-Squares* method. In the backward pass, the error signals propagate backwards and the premise parameters are updated by *Gradient Descent*.

3.6 Conclusions

This chapter describes in details the two famous rule-based fuzzy models; the linguistic fuzzy model and Takagi-Sugeno fuzzy model. Fuzzy models are the most important modeling tools based on fuzzy set theory. Conventional fuzzy inference systems are typically built by domain experts and having their drawbacks once a controller is constructed.

T-S fuzzy modeling paradigm is important issue in this chapter, since they represent the methodology of nonlinear modeling. Hence, the chapter has indeed presented adequate issues regarding the T-S models as a quasi-linear system. As will be shown later in chapter four, T-S fuzzy models is used to model nonlinear dynamic systems resulting in a state space sub-models of the linearized system.

Back-propagation learning rule, in addition to the ANFIS have been presented as the building blocks of the T-S modeling technique. Finally the chapter has ended with an emphasizes on the ANFIS learning algorithm, learning the leader prepared to the T-S modeling of nonlinear systems that will be presented in chapter four.

Chapter Four:

Takagi-Sugeno Fuzzy Clustering and Modeling of Nonlinear Systems

4.1 Introduction

Developing mathematical models of real systems is a central topic in many disciplines of engineering and science. Models can be used for simulations, analysis of the system's behavior, better understanding of the underlying mechanisms in the system, design of new processes, or design of controllers.

Takagi-Sugeno T-S modeling plays an essential role in deriving local linear models of the nonlinear dynamic system under concern. Through the use of the heuristic rules inherent in the fuzzy systems, T-S fuzzy models, then makes it possible to have a transparent like system which is governed by the fuzzy inference system and rules.

T-S fuzzy models have indeed received a focused attention in terms of their utilization in most advanced control paradigms. In a similar way fuzzy clustering has been utilized as well in classifying the data-driven fuzzy modeling, since it draws a methodology for assigning label like to similar data. Such assignment does gives quantitative directions for shaping the fuzzy membership functions.

T-S fuzzy models will then be utilized to design H^∞ robust controllers for nonlinear systems. Such H^∞ robust controllers will then be selected based on the operating condition of the system under control.

Once T-S fuzzy linear models are obtained, their state space models are computed, since most advanced control methodologies depend on the state space system model.

Model validation and verification is also an important task in the modeling paradigm. This is due to the choice of the right model from a number of models that might present similar characteristics. Statistically validated models in addition to probabilistic validation are used sometimes to make the suitable choice of a system model.

This chapter divided into eleven sections integrated together to clarify the idea of T-S modeling of nonlinear dynamic systems. Beginning with an overview of modeling terminology in section two. Then, discussing fuzzy modeling of dynamic systems and how to build such fuzzy models in section four and five. Section six introduces neurofuzzy modeling, the structural and parametric tuning in such neurofuzzy system. The concept of modeling based on fuzzy clustering is presented in section seven and the concept of extraction of a state space model from T-S sub-models is explained in section eight. Section nine

discusses the validation of the resulted fuzzy models. Finally, T-S modeling is applied to two case studies of different dimensions showing its strength and characteristics.

4.2 Overview of Modeling Terminology:

WHITE BOX MODELING

Traditionally, modeling is seen as a conjunction of a thorough understanding of the system's nature and behavior, and of a suitable mathematical treatment that leads to a usable model. This approach is usually termed "*white-box*" (physical, mechanistic, first-principle) modeling. However, the requirement for a good understanding of the physical background of the problem at hand proves to be a *severe limiting factor* in practice, when complex and poorly understood systems are considered. Difficulties encountered in conventional white-box modeling can arise, for instance, from poor understanding of the underlying phenomena, inaccurate values of various process parameters, or from the complexity of the resulting model. A complete understanding of the underlying mechanisms is virtually impossible for a majority of real systems.

However, gathering an acceptable degree of knowledge needed for physical modeling may be a very difficult, time-consuming and expensive or even impossible task. Even if the structure of the model is determined, a major problem of obtaining accurate values for the parameters remains. It is the task of system identification to estimate the parameters from data measured on the system. Identification methods are currently developed to a mature level for linear systems only. Most real processes are, however, nonlinear and can be approximated by linear local models only.

BLACK BOX MODELING

Another approach assumes that the system under study can be approximated using some sufficiently general "*black-box*" structure used as a general function approximator. The modeling problem then reduces to postulating an appropriate structure of the approximator, in order to correctly capture the dynamics and nonlinearity of the system. In black-box modeling, the structure of the model is hardly related to the structure of the real system. The identification problem consists of estimating the parameters of the model. If representative process data are available, black-box models usually can be developed quite easily, without requiring process-specific knowledge. A severe drawback of this approach is that the structure and parameters of these models usually do not have any physical significance. Such models cannot be used for analyzing a system's behavior otherwise than by numerical simulation. In

addition they cannot be scaled up or down when moving from one process scale to another, and therefore are less useful for industrial practice.

GRAY BOX MODELING

There is a range of modeling techniques that attempt to combine the advantages of the white-box and black-box approaches. In this way, the known parts of the system are modeled using physical knowledge, and the unknown or less certain parts are approximated in a black-box manner, using process data and black-box modeling structures with suitable approximation properties. These methods are often denoted as hybrid or “gray-box” modeling.

INTELLIGENT MODELING

A common drawback of most standard modeling approaches is that they cannot make effective use of extra information, such as the knowledge and experience of engineers and operators, which is often imprecise and qualitative in its nature. The fact that humans are often able to manage complex tasks under significant uncertainty has stimulated the search for alternative modeling and control paradigms. In the case of “intelligent” modeling and control methodologies, which employ techniques motivated by biological systems and human intelligence to develop models and controllers for dynamic systems, have been introduced.

These techniques explore alternative representation schemes, using, for instance, natural language, rules, semantic networks or qualitative models, and possess formal methods to incorporate extra relevant information. Fuzzy modeling and control are typical examples of techniques that make use of *human knowledge* and deductive processes. The different modeling paradigms are summarized in Table [4.1].

Modeling approach	Source of information	Method of acquisition	Example	Deficiency
Mechanistic (white-box)	formal knowledge and data	mathematical (Lagrange Equ.)	differential equations	cannot use “soft” knowledge
Black-box	data	optimization (learning)	regression, neural network	cannot at all use knowledge
Fuzzy	various knowledge and data	knowledge-based + learning	rule-based model	“curse” of dimensionality

Table [4.1] Different modeling paradigms.

4.3 Practical Relevance of Fuzzy Modeling

There are a number of practical issues that are related to the fuzzy modeling, some of these issues are stated below:

(i) INCOMPLETE OR VAGUE KNOWLEDGE ABOUT SYSTEMS.

Conventional system theories rely on crisp mathematical models of systems, such as algebraic and differential or difference equations. For a large number of practical problems, gathering of an acceptable degree of knowledge needed for physical modeling is a difficult, time-consuming and expensive or even impossible task. In the majority of systems, the underlying phenomena are understood only partially and crisp mathematical models cannot be derived or are too complex to be useful. A significant portion of information about some systems is available as the knowledge of human experts, process operators and designers. This knowledge may be vague and uncertain to be expressed by mathematical functions. It is, however, often possible to describe the functioning of systems by means of natural language, in the form of IF-THEN rules.

(ii) ADEQUATE PROCESSING OF IMPRECISE INFORMATION.

Precise numerical computation with conventional mathematical models only makes sense when the parameters and input data are accurately known. As this is often not the case, a modeling framework is needed which can adequately process not only the given data, however, also the associated uncertainty. Stochastic approach is a traditional way of dealing with uncertainty, where it has been recognized that not all types of uncertainty can be dealt within the stochastic framework. Various alternative approaches have been proposed, *Fuzzy Logic* and *Set Theory* being one of them.

(iii) TRANSPARENT (GRAY-BOX) MODELING AND IDENTIFICATION.

Many real-world systems are inherently nonlinear and cannot be represented by linear models used in conventional system identification. Recently, there is a strong focus on the development of methods for the identification of nonlinear systems from measured data. Artificial Neural Networks and fuzzy models belong to the most popular model structures used. From the input-output view, fuzzy systems are flexible mathematical functions, which can approximate other functions or just data (measurements) with a desired accuracy.

Compared to other well-known approximation techniques such as Artificial Neural Networks, Fuzzy Systems provide a more transparent representation of the system under study, which is mainly due to the possible linguistic interpretation in the form of rules. The logical structure

of the rules facilitates the understanding and analysis of the model in a semi-qualitative manner, close to the way human reason about the real world.

4.4 Modeling of Dynamic Systems

Before discussing fuzzy modeling, recall that time-invariant dynamic systems are in general modeled by static functions using the concept of the system's state. Given the state of a system with a given input, the next state $x(k+1)$ can be determined. In the sense of discrete-time setting, it can be written as,

$$x(k+1) = f(x(k), u(k)) \quad (4.1)$$

where $x(k)$ and $u(k)$ are the state and the input at time k , respectively, and f is a static function.

Fuzzy models of different types can be used to approximate the state-transition function. As the state of a system is often not measured, input-output modeling is usually applied. The most common is the NARX (Nonlinear AutoRegressive with eXogenous input) model, which is given by,

$$y(k+1) = f(y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-n_u+1)) \quad (4.2)$$

where $y(k), \dots, y(k-n_y+1)$ and $u(k), \dots, u(k-n_u+1)$ denote the past model outputs and inputs respectively and n_y, n_u are integers related to the model order (usually selected by the designer). For instance, a linguistic fuzzy model of a dynamic system may consist of rules of the following form,

$$\begin{aligned} R_i: & \text{ IF } y(k) \text{ is } A_{i1} \text{ AND } y(k-1) \text{ is } A_{i2} \text{ AND, } \dots, y(k-n+1) \text{ is } A_{in} \text{ AND } u(k) \text{ is } B_{i1} \\ & \text{ AND } u(k-1) \text{ is } B_{i2} \text{ AND, } \dots, u(k-m+1) \text{ is } B_{im} \text{ THEN } y(k+1) \text{ is } C_i \end{aligned} \quad (4.3)$$

The dynamic behavior is taken care of through an external dynamic filters added to the fuzzy system shown in Figure [4.1]. In Equ. [4.3], the input dynamic filter is a simple generator of the lagged inputs and outputs, and no output filter is used.

Since the fuzzy models can approximate any smooth function to any degree of accuracy [35], models of the type in Equ. [4.3] can approximate any observable and controllable modes of a large class of discrete-time nonlinear systems.

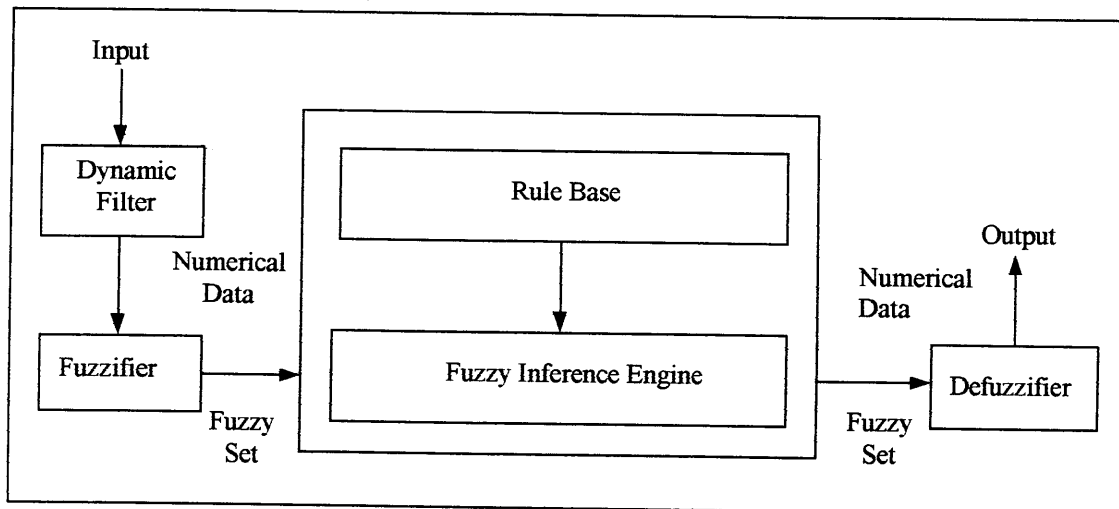


Figure [4.1] A generic fuzzy system with fuzzification and defuzzification

4.5 Building Fuzzy Models

Two common sources of information for building fuzzy models are the prior knowledge and data (process measurements). Prior knowledge can be of a rather approximate nature (qualitative knowledge, heuristics), which usually originates from “experts”, i.e., system operators.

Building fuzzy models from data involves methods based on fuzzy logic and approximate reasoning, in addition to ideas originating from the field of neural networks, data analysis and conventional systems identification.

4.5.1 Structure and Parameters

The design of fuzzy models requires two basic items: the *structure* and the *parameters* of the model. The structure determines the flexibility of the model in approximation (unknown) mappings. The parameters are then tuned (estimated) to fit the data at hand. A model with a rich structure is able to approximate more complicated functions, but, at the same time, has worse generalization properties.

In fuzzy models, structure selection involves the following choices:

- Input and output variables.

With complex systems, it is not always clear which variables should be used as inputs to the model. In the case of dynamic systems, one also must estimate the order of the system. For the input-output NARX model (Equ. [4.2]), i.e. to define the number of input and output lags n_y and n_u , respectively. Prior knowledge, insight in the process behavior and the purpose of modeling are the typical sources of information for this choice. Sometimes, automatic data-driven selection can be used to compare different choices in terms of some performance criteria.

- Structure of the rules.

This choice involves the model type (linguistic, singleton, T-S) and the antecedent form. Important aspects are the purpose of modeling and the type available knowledge.

- Number and type of membership functions for each variable.

This choice determines the level of detail (granularity) of the model. The purpose of modeling and details of available knowledge will influence this choice. Automated, data-driven methods can be used to add or remove membership functions from the model.

- Type of the inference mechanism, connective operators, defuzzification method.

These choices are restricted by the type of fuzzy model (Mamdani, T-S). Within these restrictions, however, some freedom remains, e.g., as to the choice of the conjunction operators. To facilitate data-driven optimization of fuzzy models (learning), differentiable operators (product, sum) are often preferred to the standard *min* and *max* operators.

Once the structure is fixed, the performance of a fuzzy model can be fine-tuned by adjusting its parameters. Tunable parameters of linguistic models are the parameters of antecedent and consequent membership functions (determine their shape and position) and the rules (determine the mapping between the antecedent and consequent fuzzy regions).

4.5.2 Knowledge-based design

To design a (linguistic) fuzzy model based on available expert knowledge, the following steps are to be conducted:

1. Select input and output variables, structure of the rules and inference, and defuzzification methods.
2. Decide on the number of linguistic terms for each variable and define the corresponding membership functions.
3. Formulate the available knowledge in terms of fuzzy IF-THEN rules.
4. Validate the model.

4.5.3 Data-Driven Acquisition/Tuning of Fuzzy Models

It is assumed that a set of N input-output data pairs $\{(\mathbf{x}_i, y_i) \mid i = 1, 2, \dots, N\}$ is available.

Recall that $\mathbf{x}_i \in \mathbb{R}^{F_{inp}}$ are input vectors and y_i are output scalars. Denote $\mathbf{X} \in \mathbb{R}^{N \times F_{inp}}$ a matrix having the vectors $\mathbf{x}_k^T$ in its rows, and $\mathbf{y} \in \mathbb{R}^N$ a vector containing the outputs y_k :

$$\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N]^T, \quad \mathbf{y} = [y_1 \dots y_N]^T \quad (4.4)$$

LEAST-SQUARES ESTIMATION OF CONSEQUENTS

The defuzzification formula of the T-S model as was explained in chapter three by Equ. [3.22], is linear in the consequent parameters, $\mathbf{a}_i, b_i$. These parameters can be estimated from the available data by least-squares techniques. Denote $\Gamma_i \in \mathbb{R}^{N \times N}$ the diagonal matrix having the normalized membership degree $h_i(\mathbf{x}_k)$ of (Equ. [3.23]) as its k^{th} diagonal element. By appending a unitary column to $\mathbf{X}$, an extended matrix $\mathbf{X}_e = [\mathbf{X}, \mathbf{1}]$ is created. Denoting $\mathbf{X}_{ie}$ the matrix in $\mathbb{R}^{N \times KN}$ composed of the products of matrices Γ_i and $\mathbf{X}_e$

$$\mathbf{X}_{ie} = [\Gamma_1 \mathbf{X}_e \ \Gamma_2 \mathbf{X}_e \ \dots \ \Gamma_K \mathbf{X}_e]. \quad (4.5)$$

The consequent parameters $\mathbf{a}_i$ and b_i are lumped into a single parameter vector θ

$$\begin{aligned} \theta &\in \Re^{K(F_{inp}+1)} \\ \theta &= [\mathbf{a}_1^T \ b_1 \quad \mathbf{a}_2^T \ b_2 \ \dots \ \mathbf{a}_K^T \ b_K]^T \end{aligned} \quad (4.6)$$

Given the data set $\mathbf{X}$, $\mathbf{y}$, now Equ. [3.22] can be written in a matrix form, $\mathbf{y} = \mathbf{X}_{ie} \theta + \varepsilon$.

This set of equations are solved for the parameter θ by:

$$\theta = [(\mathbf{X}_{ie})^T \mathbf{X}_{ie}]^{-1} (\mathbf{X}_{ie})^T \mathbf{y}. \quad (4.7)$$

This is an optimal least-squares solution which gives the minimal prediction error. In addition, it may bias the estimates of the consequent parameters as parameters of the local models. If an accurate estimate of a local model parameters is desired, a weighted least-squares approach applied per rule as given by,

$$[\mathbf{a}_i^T, b_i]^T = [\mathbf{X}_e^T \Gamma_i \mathbf{X}_e]^{-1} \mathbf{X}_e^T \Gamma_i \mathbf{y}. \quad (4.8)$$

In this case, the consequent parameters of individual rules are estimated independently of each other, and therefore are not “biased” by the interactions of the rules. By omitting $\mathbf{a}_i$ for all $1 \leq i \leq K$, Eqs. [4.7] and [4.8] are directly applied to the singleton model.

FROM FUZZY TO NEUROFUZZY

In conventional fuzzy approach, *membership functions* and *consequent models* are fixed by the model designer according to a prior knowledge. If this knowledge is not available but a set of input-output data is observed from the system, the components of the fuzzy system (membership functions and consequent models) can be represented in a parametric form and the parameters are tuned by a learning procedure. In this case the fuzzy system turns into a neurofuzzy approximation system. Neurofuzzy systems are a powerful trade off in terms of readability and efficiency between a human-like representation of the model and a fast learning method. However, what mainly distinguishes neurofuzzy estimates from other kinds of nonlinear approximators is their potentiality for combining available a priori first principle models with data driven modeling techniques. The learning methods provide the adaptation of the inference system and fuzzy architecture allows an easy integration into the system of available knowledge about the process to be modeled.

4.6 NeuroFuzzy Modeling

In Section 4.5.3 we have looked at the parameters that are linearly related to the output can be (optimally) estimated by least-squares methods. In order to optimize also the parameters which are related to the output in a nonlinear way, training algorithms known from the area of neural networks can be employed. At the computational level, a fuzzy model can be seen as a layered structure (network), similar to artificial neural networks. Hence, this approach is usually referred to as neurofuzzy modeling [1], [2]. Figure [4.2] shows an example of a singleton fuzzy model with two rules represented as a network. The rules are:

$$\begin{aligned} \text{IF } x_1 \text{ is } A_{11} \text{ AND } x_2 \text{ is } A_{21} \text{ THEN } y=b_1 \\ \text{IF } x_1 \text{ is } A_{12} \text{ AND } x_2 \text{ is } A_{22} \text{ THEN } y=b_2 \end{aligned}$$

The nodes in the first layer compute the membership degree of the inputs in the antecedent fuzzy sets. The product nodes $\prod$ in the second layer represent the antecedent conjunction operator. The normalization node N and the summation node $\sum$ realize the fuzzy-mean operator (Equ. [3.22]).

Using smooth antecedent membership functions, such as a Gaussian function, as given below in Equ. [4.9]:

$$\mu_{A_{ij}}(x_j, c_{ij}, \tau_{ij}) = \exp\left(-\left(\frac{x_j - c_{ij}}{2\tau_{ij}}\right)^2\right), \quad (4.9)$$

in which c_{ij} and τ_{ij} parameters are adjusted by gradient-descent learning algorithms, such as back-propagation. This allows for a fine-tuning of the fuzzy model to the available data in order to optimize its prediction accuracy.

STRUCTURAL AND PARAMETRIC LEARNING IN NEUROFUZZY INFERENCE SYSTEMS

In a neurofuzzy system, two types of tuning are required, namely structural and parametric tuning. Structural tuning aims to find a suitable number of rules and a proper partition of the input space. Once available a satisfactory structure, the parametric tuning searches for the optimal membership functions together with the optimal parameters of the consequent models. There may be a lot of structure/parameter combinations which make the fuzzy model behave in a satisfactory way. The problem can be formulated as that of finding the structure complexity which will give the best performance in generalization. In our approach we

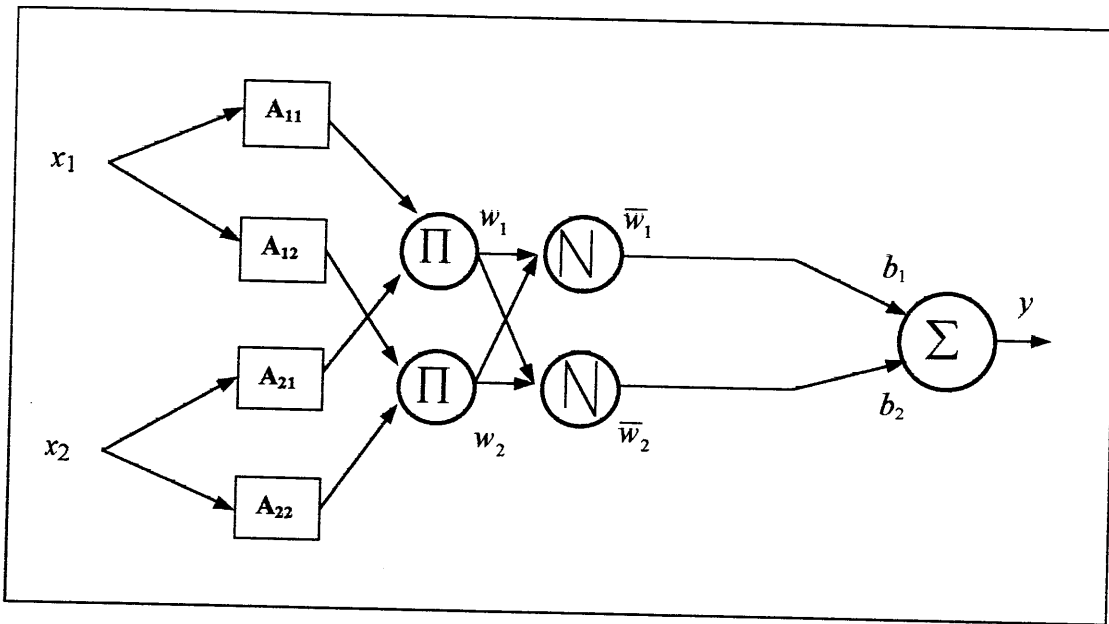


Figure [4.2] Neurofuzzy network architecture with two rules.

choose the number of rules as the measure of complexity to be properly tuned on the basis of available data. We adopt an incremental approach where different architectures having different complexity (i.e. number of rules) are first assessed in cross-validation and then compared in order to select the best one. The whole learning procedure is represented in the flow chart in Figure [4.3].

The initialization of the architecture is provided by a hyper-ellipsoid fuzzy clustering procedure inspired by Babuska and Verbruggen [36]. This procedure clustering the data in the input-output domain obtaining a set of hyper-ellipsoids which are a preliminary rough representation of the input/output mapping. Methods for initializing the parameters of a fuzzy inference system form the outcome of the fuzzy clustering procedure. Here we use the axes of the ellipsoids (eigenvectors of the scatter matrix) to initialize the parameters of the consequent functions, we project the cluster on the input domain to initialize the centers of the antecedents and we adopt the scatter matrix to compute the width of the membership functions.

Once the initialization is done, the learning procedure begins. Two optimization loops are nested: the parametric and the structural one. The parametric loop (the inner one) searches for the best set of parameters by minimizing a sum-of-squares cost function which depends exclusively on the training set. In the case of linear T-S models this minimization procedure can be decomposed in a least-squares problem to estimate the linear parameter of the consequent models and a nonlinear minimization to find the parameters of the membership functions.

The structural identification loop (the outer one) searches for the best structure, in terms of optimal number of rules, by increasing gradually the number of local models.

4.7 Fuzzy Clustering

BASICS OF FUZZY CLUSTERING

Identification methods based on fuzzy clustering originate from data analysis and pattern recognition, where the concept of graded membership is employed to represent the degree to which a given object, represented as a vector of features, is similar to some prototypical object. Degree of similarity can be calculated using a suitable distance measure. Based on that similarity, feature vectors can be clustered such that vectors within a cluster are as similar as possible, and vectors from different clusters are as dissimilar as possible.

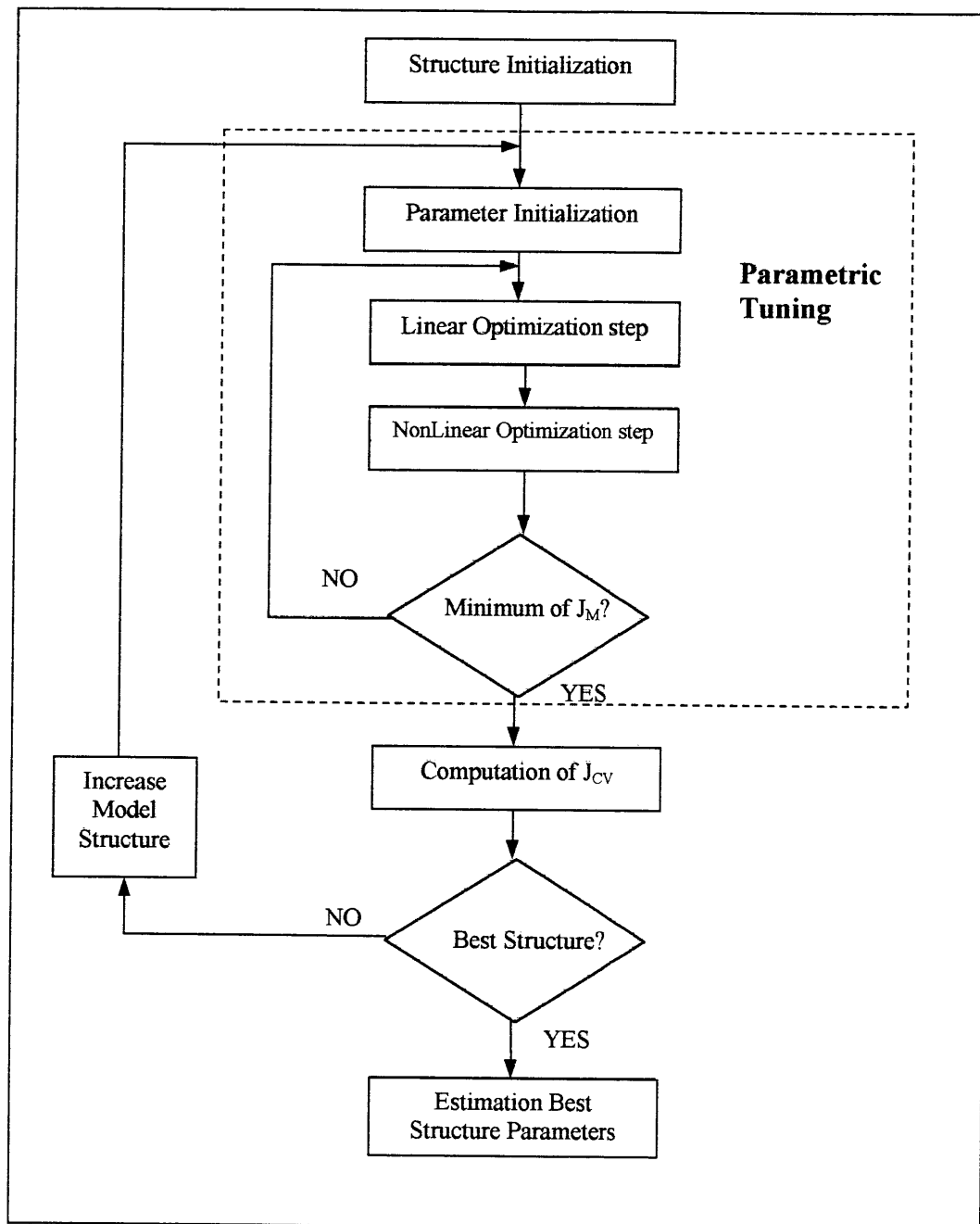


Figure [4.3] Flow-chart of the neurofuzzy learning procedure.

This idea of fuzzy clustering is depicted in Figure [4.4a]. Data is clustered into two groups with *prototypes* v_1 and v_2 , using the Euclidean distance measure. The partitioning of the data is expressed in the fuzzy partition matrix whose elements μ_{ij} are degrees of membership of each data points $[x_i, y_i]$ in a fuzzy cluster with prototypes v_j .

Fuzzy IF-THEN rules can be extracted by projecting the clusters onto the axes. Figure [4.4a] shows a data set with two apparent clusters and two associated fuzzy rules. The concept of similarity of data to a given prototype leaves enough space for the choice of an appropriate distance measure and of the character of the prototype itself. For example, prototypes can be defined as linear subspaces, or the clusters can be ellipsoids with adaptively determined shape [37], as shown in Figure [4.4b]. From these clusters, the antecedent membership functions and the consequent parameters of the T-S model can be extracted [38] as follows:

$$\begin{array}{ll} \text{IF } x \text{ is } A_1 & \text{THEN } y = a_1x + b_1 \\ \text{IF } x \text{ is } A_2 & \text{THEN } y = a_2x + b_2 \end{array}$$

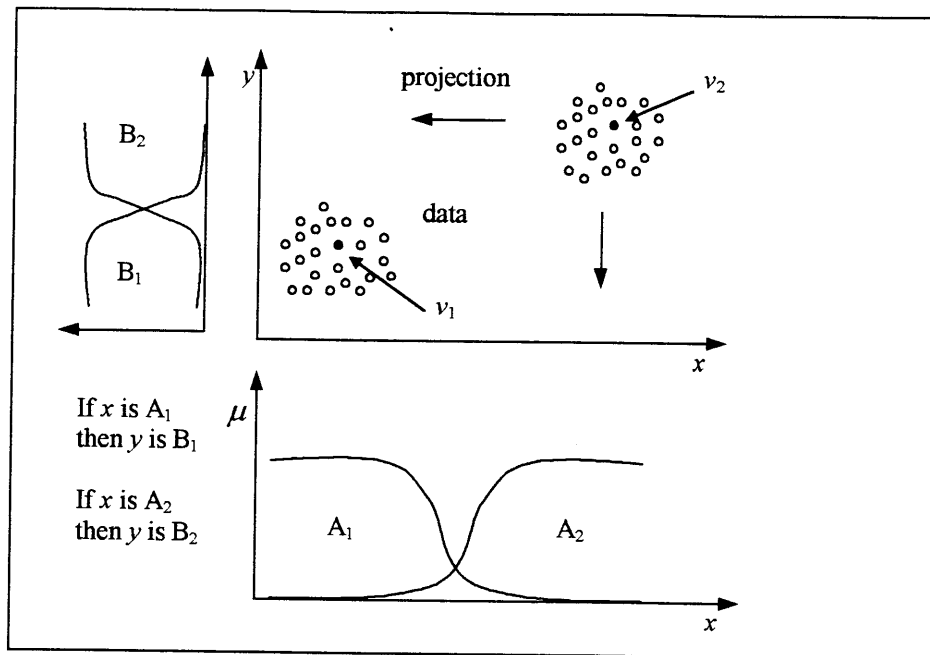
Each obtained cluster is represented by one rule in the T-S model. Membership functions for fuzzy sets A_1 and A_2 are generated by point-wise projection of the partition matrix onto the antecedent variables. Such point-wise defined fuzzy sets are then approximated by a suitable parametric function. The consequent parameters for each rule are obtained as least squares estimates (Equ. [4.8]).

FUZZY CLUSTERING ALGORITHM

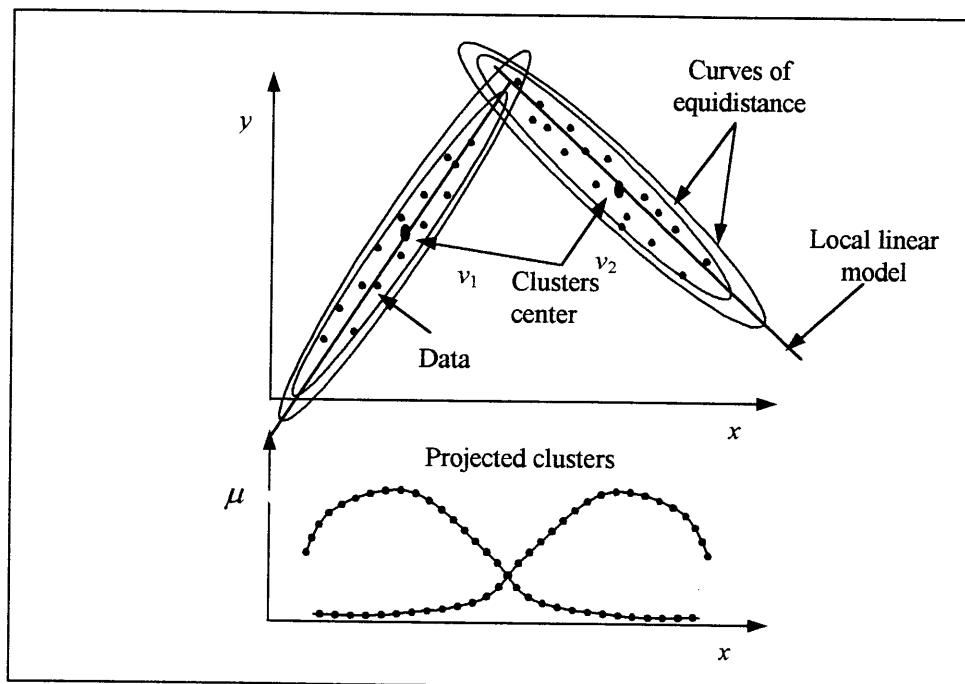
Consider a finite set of elements $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ as being elements of the F_{inp} dimensional Euclidean space $\mathcal{R}^{F_{inp}}$, that is, $x_j \in \mathcal{R}^{F_{inp}}$, $j = 1, 2, \dots, n$. The issue is to perform a partition of such collection of elements into c fuzzy sets with respect to a given criterion. Hereby c is a given number of clusters. The criterion is usually to optimize an objective function that acts as a performance index of clustering. The end result of fuzzy clustering can be expressed by a partition matrix U such that,

$$U = [u_{ij}]_i \quad i = 1, \dots, c. \quad j = 1, \dots, n \quad (4.10)$$

In Equ. [4.10], u_{ij} is a numerical value in $[0, 1]$ and expresses the degree to which an element x_j belongs to the i^{th} cluster. However, there are two additional constraints on the value of u_{ij} . First, a total membership of the element $x_j \in X$ in all classes is equal to unity; that is,



(a) Rule based interpretation of fuzzy clusters



(b) Hyperellipsoidal fuzzy clusters

Figure [4.4] Identification by fuzzy clusters.

$$\sum_{i=1}^c u_{ij} = 1 \quad \text{for all } j=1,2, \dots, n \quad (4.11)$$

Second, every constructed cluster is a nonempty and different from the entire set; that is,

$$0 < \sum_{j=1}^n u_{ij} < n \quad \text{for all } i=1,2, \dots, c \quad (4.12)$$

The general form of the objective function used in fuzzy clustering is,

$$J(u_{ij}, v_k) = \sum_{i=1}^c \sum_{j=1}^n \sum_{k=1}^c g[w(x_i), u_{ij}] d(x_j, v_k) \quad (4.13)$$

where $w(x_i)$ is a prior weight for each x_i and $d(x_j, v_k)$ is the degree of dissimilarity between the data x_j and the supplemental element v_k , which can be considered the central vector of the k^{th} cluster. Degree of dissimilarity is defined as a measure that satisfies two assumptions given by,

$$(i) \quad d(x_j, v_k) \geq 0, \quad (4.14)$$

$$(ii) \quad d(x_j, v_k) = d(v_k, x_j) \quad (4.15)$$

Based on the above background, fuzzy clustering can be precisely formulated as an optimization problem:

Minimize

$$J(u_{ij}, v_k) = \sum_{i=1}^c \sum_{j=1}^n \sum_{k=1}^c g[w(x_i), u_{ij}] d(x_j, v_k), \quad i, k = 1, 2, \dots, c; j = 1, 2, \dots, n \quad (4.16)$$

Subject to

$$\sum_{i=1}^c u_{ij} = 1 \quad \text{for all } j=1,2, \dots, n$$

$$0 < \sum_{j=1}^n u_{ij} < n \quad \text{for all } i=1,2, \dots, c$$

One of the widely employed clustering methods based on Equ. [4.16] is the fuzzy c -means (FCM) algorithm. The objective function of the FCM algorithm is expressed in the form of,

$$J(u_{ij}, v_k) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|x_j - v_i\|^2, \quad m > 1 \quad (4.17)$$

where m is called exponential weight that influences the degree of fuzziness of the membership (partition) matrix. To solve this minimization problem, the objective function $J(u_{ij}, v_k)$ is differentiated in Equ. [4.17] with respect to v_i (for fixed u_{ij} , $i=1, \dots, c$, $j=1, \dots, n$) and to u_{ij} (for fixed v_i , $i=1, \dots, c$) and apply the conditions of Equ. [4.11], obtaining,

$$v_i = \frac{1}{\sum_{j=1}^n (u_{ij})^m} \sum_{j=1}^n (u_{ij})^m x_j, \quad i = 1, 2, \dots, c \quad (4.18)$$

$$u_{ij} = \frac{\left(1/\|x_j - v_i\|^2\right)^{\frac{1}{m-1}}}{\sum_{k=1}^c \left(1/\|x_j - v_k\|^2\right)^{\frac{1}{m-1}}}, \quad i = 1, 2, \dots, c; \quad j = 1, 2, \dots, n \quad (4.19)$$

The system described by the Eqs. [4.18] and [4.19] cannot be solved analytically. However, the FCM algorithm provides an iterative approach to approximating the minimum of the objective function starting from a given position. This algorithm is summarized by the following:

ALGORITHM FCM: FUZZY C-MEAN ALGORITHM

STEP 1: Select a number of clusters c ($2 \leq c \leq n$) and exponential weight m ($1 < m < \infty$). Choosing an initial partition matrix $U^{(0)}$ and a termination criterion ϵ . Set the iteration index l to 0.

STEP 2: Calculate the fuzzy cluster centers $\{v_i^{(l)} | i = 1, 2, \dots, c\}$ by using $U^{(l)}$ and Equ. [4.18]

STEP 3: Calculate the new partition matrix $U^{(l+1)}$ by using $\{v_i^{(l)} | i = 1, 2, \dots, c\}$ and Equ. [4.19].

STEP 4: Evaluate Δ , where $\Delta = \|U^{(l+1)} - U^{(l)}\| = \max_{i,j} |u_{ij}^{(l+1)} - u_{ij}^{(l)}|$. If $\Delta > \epsilon$, then set $l=l+1$ and go to step 2. If $\Delta \leq \epsilon$, then stop.

END FCM

The iterative procedure described above minimizes the objective function in Equ. [4.17] and leads to any of its local minima.

4.8 Extraction of a Linear State Space Model

At each sample time k , given an operating point condition (for example $u(k-1)$ and $y(k-1)$), a local linear fuzzy state-space model can be constructed via calculating the degree of fulfillment $\mu_i(x(k))$ of the antecedents, using product as the fuzzy logic AND operator. The inference gives:

$$y_l(k+1) = \frac{\sum_{i=1}^r \mu_{li}(x_l(k)) y_{li}(k+1)}{\sum_{i=1}^r \mu_{li}(x_l(k))} \quad (4.20)$$

$$y_{li}(k+1) = (\zeta_{li} y(k) + \eta_{li} u(k) + \theta_{li}) \quad (4.21)$$

Defining ζ_l , η_l and θ_l as follows:

$$\zeta_l = \frac{\sum_{i=1}^r \mu_{li}(x_l(k)) \zeta_{li}}{\sum_{i=1}^r \mu_{li}(x_l(k))} \quad (4.22)$$

$$\eta_l = \frac{\sum_{i=1}^r \mu_{li}(x_l(k)) \eta_{li}}{\sum_{i=1}^r \mu_{li}(x_l(k))} \quad (4.23)$$

$$\theta_l = \frac{\sum_{i=1}^r \mu_{li}(x_l(k)) \theta_{li}}{\sum_{i=1}^r \mu_{li}(x_l(k))} \quad (4.24)$$

Defining $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{y}$ for the state-space description as,

$$\mathbf{x}(k) = [x_1(k) \dots x_1(k - n_{y1}) \dots x_{n_o}(k) \dots x_{n_o}(k - n_{yn_o}) \quad u_1(k-1) \dots u_{n_i}(k - n_d - n_u)]^T \quad (4.25)$$

$$\mathbf{u}(k) = [u_1(k) \quad u_2(k) \dots u_{n_i}(k)]^T \quad (4.26)$$

$$\mathbf{y}(k) = [x_1(k) \quad x_2(k) \dots x_{n_i}(k)]^T \quad (4.27)$$

In order to employ Quadratic Programming for systems which depend on current as well as on the previous inputs, it is necessary to construct a state-space representation, such that the state vector $\mathbf{x}(t)$ to accommodate not only the state variables, appearing in $\mathbf{y}(k)$, but also the previous inputs and the offset as last element. This results in a system with only current

inputs, but leads to a more complex $\mathbf{A}$ -matrix. The latter contains also η s, corresponding to the previous inputs.

If the maximal delay in the input i , $i=1, \dots, n_i$ is $u_{i,dmax}$, then the number of the additional columns is $\sum_{i=1}^{n_i} \max(u_{i,dmax} - 1, 0)$. In the last column of $\mathbf{A}$ are stored the offsets θ s.

The columns with η s corresponds to the previous inputs, stored in the state vector; these columns are not included in $\mathbf{B}$. The ones in $\mathbf{A}$ correspond to the delayed values of a certain variable. The local linear system matrices are derived as follows:

$\mathbf{A}$ is $\alpha \times \alpha$ square matrix, where

$$\alpha = \alpha_1 + \sum_{i=1}^{n_i} \max(u_{i,dmax} - 1, 0) + 1, \alpha_1 = \sum_{j=1}^{n_{oi}} \max(n_{yj} - 1),$$

$\mathbf{B}$ is an $\alpha \times n_i$ and $\mathbf{C}$ is a $n_o \times \alpha$ matrix:

$$\mathbf{A} = \begin{bmatrix} \zeta_{1,1} & \zeta_{1,2} & \dots & \dots & \dots & \dots & \zeta_{1,\alpha_1} & \eta_{1,i} & \dots & \eta_{1,j} & \theta_{1,1} \\ 1 & 0 & 0 & \dots & \dots & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \vdots & \ddots & \dots & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & \dots & \ddots & \ddots & \dots & \dots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \zeta_{2,1} & \zeta_{2,2} & \dots & \dots & \dots & \dots & \zeta_{2,\alpha_1} & \eta_{2,i} & \dots & \eta_{2,j} & \theta_{2,1} \\ 0 & \vdots & \ddots & \ddots & \dots & \dots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \zeta_{n_o,1} & \zeta_{n_o,2} & \dots & \dots & \dots & \dots & \zeta_{n_o,\alpha_1} & \eta_{n_o,i} & \dots & \eta_{n_o,j} & \theta_{n_o,\alpha_1} \\ 0 & \dots & 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}, \quad (4.28)$$

$$\mathbf{B} = \begin{bmatrix} \eta_{1,1} & \eta_{1,2} & \dots & \eta_{1,n_i} \\ 0 & \dots & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & \dots & 0 \\ \eta_{2,1} & \eta_{2,2} & \dots & \eta_{2,n_i} \\ \vdots & \ddots & \ddots & \vdots \\ \eta_{n_o,1} & \eta_{n_o,2} & \dots & \eta_{n_o,n_i} \\ \vdots & \ddots & \ddots & \vdots \end{bmatrix}, \quad (4.29)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & \dots & \dots & 1 & 0 & \dots & 0 \end{bmatrix}. \quad (4.30)$$

The ones in $\mathbf{C}$ are positioned such that $y_i(k) = x_i(k)$.

The obtained linear state-space model will be utilized in the robust control system discussed later in chapter five. At any time index k , initially the control signal $u(k-1)$ is used.

However, after the optimization, $u(k)$ is available and could be used in next iterations.

4.9 Fuzzy Model Validation

The importance of a model structure is clear and in this section methods which can be used to assess the performance of different models are introduced [39].

4.9.1 Validation via Hypothesis Testing:

Hypothesis testing addresses the problem of determining which of two hypotheses; the null hypothesis (H_0) and the alternative hypothesis (H_1), is more statistically likely given the data. Applied to the problem of model selection, it is generally presented as follows.

Considering a model with a parameter vector $\mathbf{w}$ composed of two disjoint sub-vectors such that,

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \end{bmatrix}$$

where $\mathbf{w}_1$ is a column vector of dimension s , $\mathbf{w}_0$ is a column vector of dimension $p-s$ and p is the dimension of $\mathbf{w}$. The two hypotheses are defined as,

$$\begin{aligned} H_0: & \quad \mathbf{w}_1 = \mathbf{z} \quad (\text{null hypothesis}) \\ H_1: & \quad \mathbf{w}_1 \neq \mathbf{z} \quad (\text{alternative hypothesis}) \end{aligned}$$

In which $\mathbf{z}$ is a zero vector. The null hypothesis represents a reduced model with some of the parameters equal to zero. Using classical hypothesis testing the aim is to determine whether there is enough evidence in the data to justify the use of the larger model.

These hypotheses are tested by using an appropriate statistic, which is simply a function of the data. Associated with any statistic (test), there are two possible errors; rejecting H_0 when

$w_1 = \mathbf{z}$ and rejecting H_1 when $w_0 \neq \mathbf{z}$, which are classified as type I and type II errors, respectively.

The aim of hypothesis testing is to find a test statistic such that the probabilities of these errors,

$$\begin{aligned}\alpha &= \text{Prob}(\text{type I error}) \\ \beta &= \text{Prob}(\text{type II error})\end{aligned}$$

are low. The probability distribution function (p.d.f) of a chosen statistic can be used to define its probabilities, α and β . Usually, the p.d.f of the statistic for the null hypothesis to be correct is split into regions; the acceptance and the critical regions. If the realization of the statistic falls in the acceptance region the null hypothesis is accepted, otherwise it is rejected. The probability α , known as the test's significance level, is the area of the critical region which is a trade-off between α and β , as α is decreased β will increase.

A common statistic for the restricted model comparison is the Log Determinate Ratio (LDR) test, where for single output modeling is expressed by:

$$d(y) = N \log \left[\frac{J_N(\hat{w}_0)}{J_N(\hat{w}_1)} \right]. \quad (4.31)$$

where J_N is the objective function. Such statistic is asymptotically equivalent to the F-test conventionally used in system identification, and converges to a $X^2(s)$ distribution as $N \rightarrow \infty$, where s is the number of degrees of freedom of this distribution. From this, given a significance level α , the acceptance and critical regions can be determined. The critical point of a $X^2(s)$ distribution, denoted by $K_\alpha(s)$, is the point where the area under the p.d.f to the right is α . If $d(y)$ is greater than $K_\alpha(s)$, there is sufficient evidence in the data to accept the null hypothesis.

Hypothesis testing as described here is very restrictive as only the comparison between two models, one being a subset of the other, is preferred.

4.9.2 Model Selection Criteria

In this section the more realistic problem of selecting a model from a large set of models is considered. There have been many different criteria derived for this purpose, selections of a model among a number of models. All these different criteria adhere to the underlying principle of parsimony which is the determination of a trade-off between model's bias (complexity) and variance (accuracy).

Hypothesis Testing

Hypothesis testing is extended to the problem of selecting a model from a number of different models, known as the model set. Given this set of competing models the best one is defined as the one that minimizes the following criteria:

$$C = N \log(J_N(\hat{w})) + pK(1). \quad (4.32)$$

the trade-off between the two terms, $\log(J_N(\hat{w}))$ (model accuracy) and p (model size), is controlled by $K(1)$ which can be set to give the desired significance level α . $K(1)$ is a parameter that defines the probability of rejecting the correct smaller model when comparing two models whose parameters differ by 1. For example if $K(1)=4$, the probability of rejecting the correct model is given by $\alpha = 0.0456$. Similar to the hypothesis test in the last section, this assumes that every model is the same structure as a full model with some of the parameters equal to zero. Despite being restrictive this is a common occurrence when models are iteratively constructed. Like most of the criteria considered in this section, this criteria is represented by the ability of the model to fit the data set $N \log(J_N(\hat{w}))$ penalized by the complexity of the model, hence explicitly enforcing the principle of parsimony.

Information Measures:

One of the original model order (i.e. $n_u + n_y$) selection criteria is derived from considering an information theoretic measure of the distance between the estimated p.d.f for the data and the true one. Using the Kullback-Leibler distance measure [50], Information measure (A_{IC}) given by,

$$A_{IC} = N \log(J_N(\hat{w})) + 2p \quad (4.33)$$

The A_{IC} can be considered a special case of the hypothesis test criteria with the significant level set by $K(1)=2$. The A_{IC} has been found by many researchers to over estimate the

model's size, this can be attributed to this small significance level. The final prediction error (F_{PE}) test, given by:

$$F_{PE} = J_N(\hat{w}) \frac{(N+p)}{(N-p)} \quad (4.34)$$

in such a way,

$$N \log(F_{PE}) = N \log \left(J_N(\hat{w}) + N \log \left[\frac{(N+p)}{(N-p)} \right] \right). \quad (4.35)$$

This loss function is asymptotically equivalent to A_{IC} as N approaches ∞ and it can be seen that it conforms to the general model criteria structure of accuracy penalized by complexity.

An alternative information theoretic approach was proposed by Rissanen [50], where the aim is to find the model that can accurately describe the data with the minimum amount of information. Based on the ideas of optimal coding theory, the Maximum Descriptor Length (M_{DL}) criteria for model selection is derived as:

$$M_{DL} = \log(J_N(\hat{w})) + \frac{p \log(N)}{N} \quad (4.36)$$

A very similar criterion can be derived from Bayesian estimation. This criteria takes the familiar structure of accuracy penalized by complexity and can be considered a hypothesis test with the significance level defined by the data set size, $\log(N)$.

Cross Validation

A very popular method of model selection is the cross validation. The empirical data set is split into two disjoint sets, one for identifying the model and one for validation.

There are several problems associated with this technique, from these:

- (i) Data is often limited, and hence to sufficiently populate the input space all data should be used during system identification;
- (ii) The separate validation data set can guide model selection to fit its own idiosyncrasies and so ideally a new independent validation set is required for each test; and
- (iii) It is computationally expensive to perform.

An alternative is to perform v -fold validation. Such a data set is split into v separate data sets D_i , where the following cross validation (C_{vv}) criteria is defined:

$$C_{vv} = \frac{1}{v} \sum_{i=1}^v J_{N_i}(\hat{w}) \quad (4.37)$$

In Equ. [4.37] $J_{N_i}(\hat{w})$ is the mean square error across D_i for the model identified on all the data except this data set. This approach requires less data, however, it is still very computationally expensive requiring v parameter identifications per model. Based on these ideas, a computationally efficient method, the Generalized Cross Validation (Gcv) has been derived:

$$G_{CV} = \frac{J_N(\hat{w})}{(1 - p/N)^2} \quad (4.38)$$

$$N \log(G_{CV}) = N \log(J_N(\hat{w})) - N \log((1 - p/N)^2). \quad (4.39)$$

This criteria exhibits similar behavior to the F_{PE} , to which it is asymptotically equivalent.

Nonlinear Models

The criterion described in the last section is only applicable to selecting models which are linear in their parameters. Much of this original work was derived for ARX type models. Nonlinear model selection has been addressed by the following criteria,

$$N_{IC} = N \log(J_N(\hat{w})) + 2p_{eff}. \quad (4.40)$$

This is equivalent to the linear hypothesis measure (and A_{IC}) where the number of parameters has been replaced by the effective number of parameters of the nonlinear model defined as,

$$p_{eff} = \text{tr} \left[\left(g^T g H^{-1} \right) \right], \quad (4.41)$$

where $\text{tr}[\cdot]$ represents the trace of the matrix and g is the variance of the gradient of the cost function defined by,

$$g = \text{Var} \left[\frac{\partial J_N(\hat{w})}{\partial w} \right] \quad (4.42)$$

and

$$H = E \left[\frac{\partial^2 J_N(\hat{w})}{\partial w^2} \right] \quad (4.43)$$

indicating expectation of the Hessian of the cost function $J_N(\cdot)$.

Both of these are estimated as averages across the data at the maximum likelihood model weights. If this effective number of parameters is much lower than the actual number, N_{IC} and the equivalent linear measure will select different models. However, due to the computation required to calculate the effective number of parameters, generally linear techniques are employed to select nonlinear models. One advantage of these measures is that they can be used to determine the amount of regularization required i.e. the value of the regularization coefficient.

In deriving many of the selection criteria above $J_N(\hat{w})$ is seen as an estimate of the noise in the data, and hence either the biased or the unbiased estimate can be used. For small data sets, this choice can be very important and the use of the unbiased estimate produces significantly better behaved selection criteria. For instance once using the unbiased estimate of a noise the A_{IC} becomes,

$$A_{IC} = N \log(J_N(\hat{w})) + 2p + N \log\left(\frac{N}{N-p}\right). \quad (4.44)$$

The difference is only significant for a small training set.

4.9.3 Correlation Tests:

Practical experience has shown that model selection criteria, described above, can potentially select incorrect models, hence further validation is required. The simplest form of model validation is by expert inspection. If the model is simple and/or can be described qualitatively, an expert can verify the model by inspection or interrogation. However, traditionally more rigorous statistical validation tests are employed in which models residuals are examined and if found to be sufficiently correlated with a function of the data then the model is inadequate.

This is achieved by defining a matrix Z , where Z is, :

$$Z(x') = [m(t), m(t-1), \dots, m(t-t_d)]^T \quad (4.45)$$

x^t is the observational vector of inputs, outputs and errors seen up to time step t i.e.

$$x^t = [u^{t-1}, y^{t-1}, e^{t-1}]^T \quad (4.46)$$

and $m(t-1)$ is a monomial of the vector x^t given by,

$$m(t) = y(t-1)u^2(t-2). \quad (4.47)$$

The following two hypothesis can be defined:

$$H_0 : e(t) \text{ is uncorrelated with } Z(x^t), E(e(t)|Z(.)) = 0$$

$$H_1 : e(t) \text{ is correlated with } Z(x^t), E(e(t)|Z(.)) \neq 0$$

where the purpose of validation is to use the data to decide if H_0 holds.

Two different test statistics have been proposed in the literature, the most common being the standard sample correlation measure $\rho(k)$, which is defined as,

$$\rho(k) = \frac{1/N \sum_{t=1}^{N-K+1} (m(t-k+1)e(t))}{\sqrt{\sum_{t=1}^{N-K+1} (m(t-k+1)m(t-k+1)) \sum_{t=1}^{N-K+1} (e(t)e(t))}} \quad (4.48)$$

where $k = 1, \dots, t_d$, and $\rho(k) \in [-1, 1]$. If H_0 holds this statistic asymptotically approaches a normal distribution, and with 95% confidence limits H_0 is accepted if $\rho(k) \in [-1.96 / \sqrt{N}, 1.96 / \sqrt{N}]$.

An alternative statistic is given by:

$$d = N \left[\sum_{t=1}^{N-t_d} (e^2(t)) \right]^{-1} \left[\sum_{t=1}^{N-t_d} (Z(t)e(t)) \right] \left[\sum_{t=1}^{N-t_d} (Z(t)Z(t)) \right]^{-1} \left[\sum_{t=1}^{N-t_d} (Z(t)e(t)) \right] \quad (4.50)$$

which is H_0 hold is asymptotically a $X^2(s)$ distribution where s is the number of delays, t_d .

For a given acceptance level (typically 95%) the critical point is found, and if elements of d are outside this acceptance region, H_0 is rejected.

4.10 Case Studies: Neurofuzzy Modeling of Nonlinear System.

The proposed neurofuzzy methodology developed in this chapter has been applied to two case studies to illustrate the respective attribute of this method. To test these algorithms further, they are applied to the modeling of two realistic nonlinear dynamical systems. Admittedly these are still only simulations but are considerably realistic, in which simulations are entirely done using MATLAB software. Two systems are considered, a nonlinear mass-spring-damper system (SISO) and a nonlinear dynamics of an antenna (MIMO) system with two coupled inputs and two coupled outputs. The two systems are described in details in appendices [A-1] and [A-2]. Hence, the objective of this section is to demonstrate the use of the neurofuzzy modeling as was illustrated by the flow chart shown in Figure [4.3].

[A] Nonlinear Spring-Mass-Damper System

Training Pattern

A block diagram was built using SIMULINK/MATLAB representing the mass-spring-damper system in order to get a 1000 sample of input/output data to be used as an identification data in the T-S fuzzy modeling technique presented in this chapter. The data was produced by applying random force signal to the input channel, with a sampling rate of 2 samples/sec. The training pattern has been selected in such a way to excite the mass-spring system and drive it over all its operating regions. The length of the training data extends up to 500 seconds. The unit of the input signal is considered as Newton, whereas the displacement were measured in meter unit. Half of the training pattern was used for the modeling whereas the other half was used to validate the fuzzy models resulted of the modeling process. For a typical sequence of inputs, the response of the mass-spring-damper system is shown in Figure [4.5].

Clustering of Training Pattern

Since the desired T-S model of the mass-spring-damper system depends on the membership centers, the training pattern has been clustered resulting in a clustered training pattern accordingly. For instance, Figure [4.6] shows the associated input-output training pattern prior to clustering whereas Figure [4.7] shows the training pattern after clustering. The clustering algorithm used in the simulation was the c-mean fuzzy clustering, and the number of clusters is chosen to be three, fuzziness parameter m was kept at 2.2 and the termination criterion $\epsilon = 0.01$.

Membership Functions and Associated Fuzzy Rules

To make the fuzzy models valid models, it is required to generate the right and suitable memberships. Hence Figure [4.8] displays the MFs of the fuzzy model that has been

extracted from the clusters. Since the mass-spring system has only two inputs (in terms of its fuzzy model), and one output, three universe of discourse are shown, where each universe of discourse has only three MFs representing the assigned three clusters. Memberships of $u(k)$ are shown to be within the limit of -4 N up to $+4$ N, representing the range of the variables. These MFs are: F_1^1, F_1^2, F_1^3 . In a similar way the fuzzification of the fuzzy variable $y(k-1)$ is also shown in addition to $y(k)$. For our case the two inputs for the fuzzy model are $u(k)$, $y(k-1)$ and the output was $y(k)$.

On the other hand, shapes of the membership functions show how the mass position $y(k)$ has been related to $y(k-1)$ and $u(k)$ via different equations representing the linear behavior of the global model (local models),

$$y(k) = [u(k), y(k-1), \Delta y(k-1)]$$

Fuzzy Model Validation

Once the fuzzy models of the nonlinear mass-spring system have been constructed, it is necessary to validate such models. As was mentioned already, there are a number of techniques that are employed to validate the constructed fuzzy model.

MEASURE OF ERROR

Figure [4.9] shows the actual mass-spring output superimposed over the fuzzy model output. From the figure, it is clear how the fuzzy model output resembles the actual system output. In this respect, the model output is able to move within 1.8 meter range of mass displacement.

Figure [4.10] shows the associated error between the two outputs, where the error is being averaged around the zero level, hence validating the fuzzy model by nearly 98% of accuracy. Further analysis of such error can be shown as well in Figure [4.11], the histogram of the error, where it is evidence how the error is being concentrated around zero with a narrow error spread.

CORRELATION FUNCTIONS

Furthermore, two approaches have been employed to investigate how the fuzzy model is strongly resembles the original system dynamics. These are the auto-correlation and the cross-correlation functions. For instance, Figure [4.12] shows the associated auto-correlation function of the modeling error signal (between fuzzy model and actual system output) as a function of lagging indices. From the figure, a good result can be seen since the correlation

lies within the limit of confidence $\pm 1.96/\sqrt{N}$ which is represented in the figure by two dashed lines. In addition to this, Figure [4.13] displays the cross-correlation of the error signal and the mass-spring input signal used for training. It shows that the two signal are not correlated.

Fuzzy Sub-models

Once the fuzzy models have been validated, it is essential to determine the associated fuzzy sub-models. Since we have been able to cluster the training pattern into three main clusters (as was shown in figure [4.7]), three fuzzy sub-models of the nonlinear mass-spring system can be constructed. By using the algorithm in section 4.8 to get T-S state space sub-models from T-S fuzzy sub-models, such models are,

$$\begin{aligned}
 \text{IF } u(k) \text{ is } F_1^1 \text{ AND } y(k-1) \text{ is } F_2^1 \quad & \text{THEN } \dot{\mathbf{x}}(t) = \mathbf{A}_1 \mathbf{x}(t) + \mathbf{B}_1 u(t) \\
 & y = \mathbf{C}_1 \mathbf{x}(t) \\
 \text{IF } u(k) \text{ is } F_1^2 \text{ AND } y(k-1) \text{ is } F_2^2 \quad & \text{THEN } \dot{\mathbf{x}}(t) = \mathbf{A}_2 \mathbf{x}(t) + \mathbf{B}_2 u(t) \\
 & y = \mathbf{C}_2 \mathbf{x}(t) \\
 \text{IF } u(k) \text{ is } F_1^3 \text{ AND } y(k-1) \text{ is } F_2^3 \quad & \text{THEN } \dot{\mathbf{x}}(t) = \mathbf{A}_3 \mathbf{x}(t) + \mathbf{B}_3 u(t) \\
 & y = \mathbf{C}_3 \mathbf{x}(t)
 \end{aligned}$$

where,

Sub-model 1:

$$\mathbf{A}_1 = \begin{bmatrix} -0.318 & -0.064 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 0.0123 \\ 0 \end{bmatrix}, \quad \mathbf{C}_1 = [1 \quad 0]$$

Sub-model 2:

$$\mathbf{A}_2 = \begin{bmatrix} -0.318 & 0.100 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0.100 \\ 0 \end{bmatrix}, \quad \mathbf{C}_2 = [1 \quad 0]$$

Sub-model 3:

$$\mathbf{A}_3 = \begin{bmatrix} -0.020 & 0.200 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B}_3 = \begin{bmatrix} 0.030 \\ 0 \end{bmatrix}, \quad \mathbf{C}_3 = [1 \quad 0]$$

These sub-models are to be employed at later stage (chapter five and six) to synthesis a robust H^∞ control system.

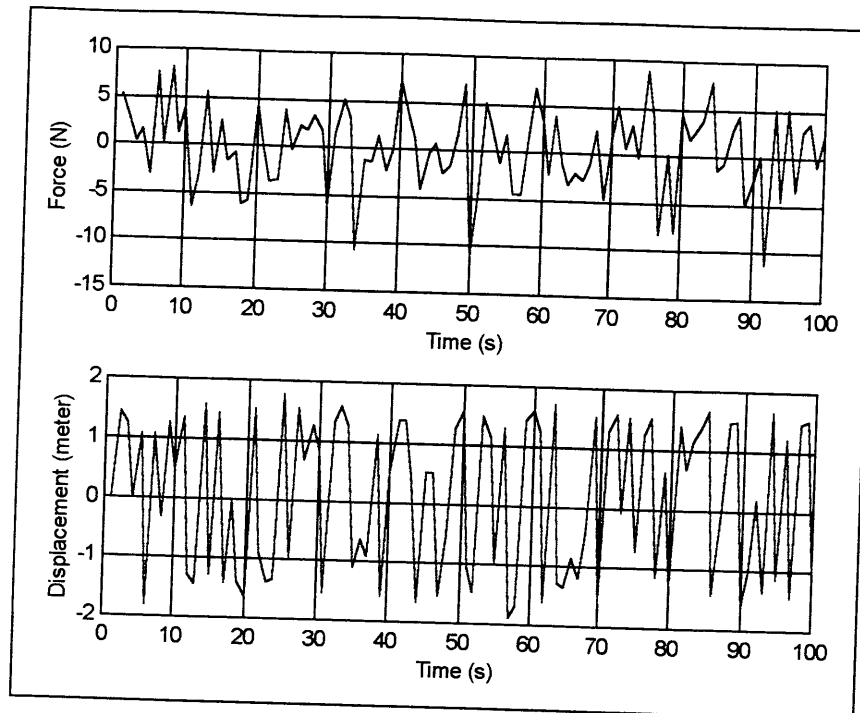


Figure [4.5] Input/output training pattern of mass-spring system.

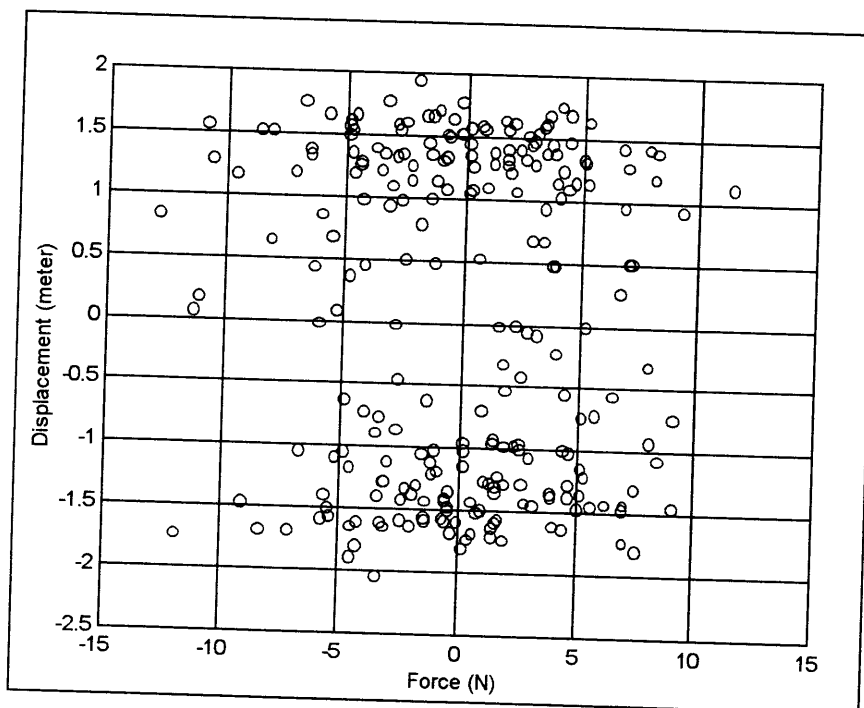


Figure [4.6] Training pattern of mass-spring system prior to clustering.

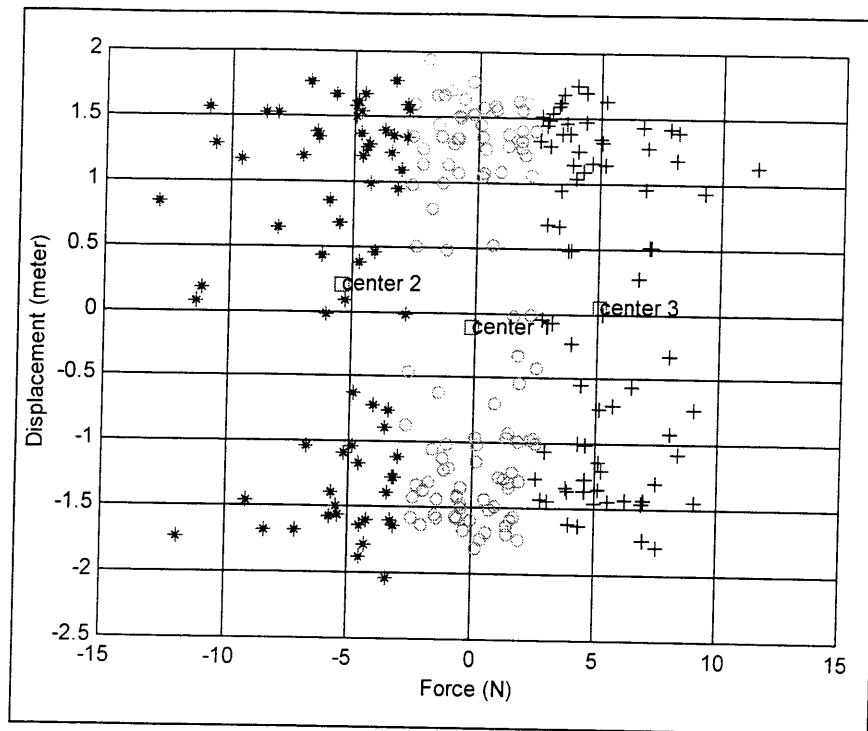


Figure [4.7] Input-output training pattern of the mass-spring system after clustering.

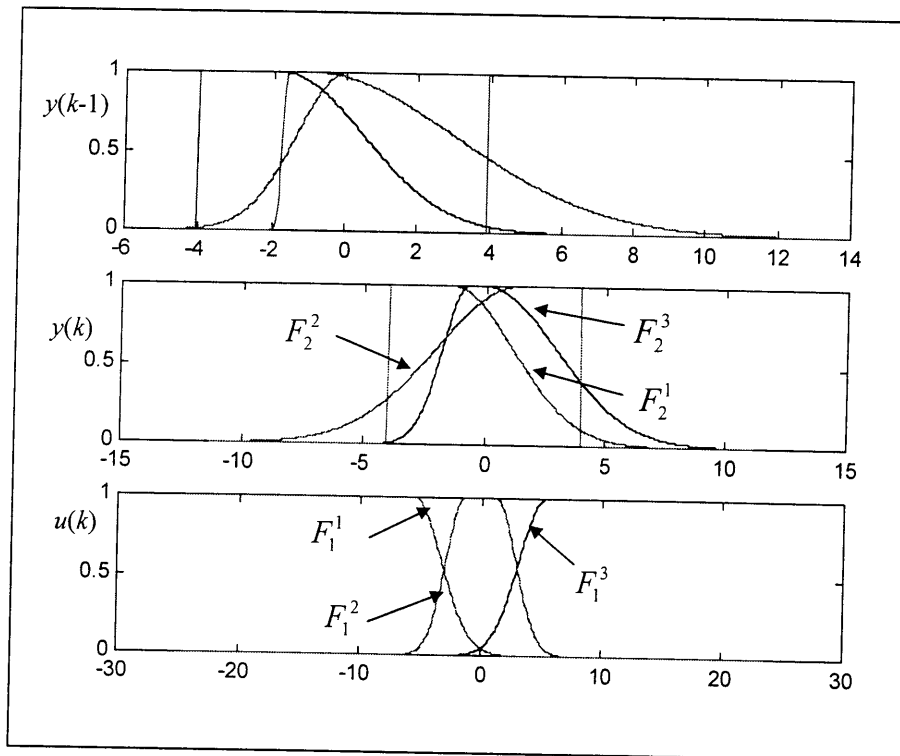


Figure [4.8] Extracted membership functions of $y(k-1)$, $u(k)$ input regressors and output $y(k)$ of the mass-spring system.

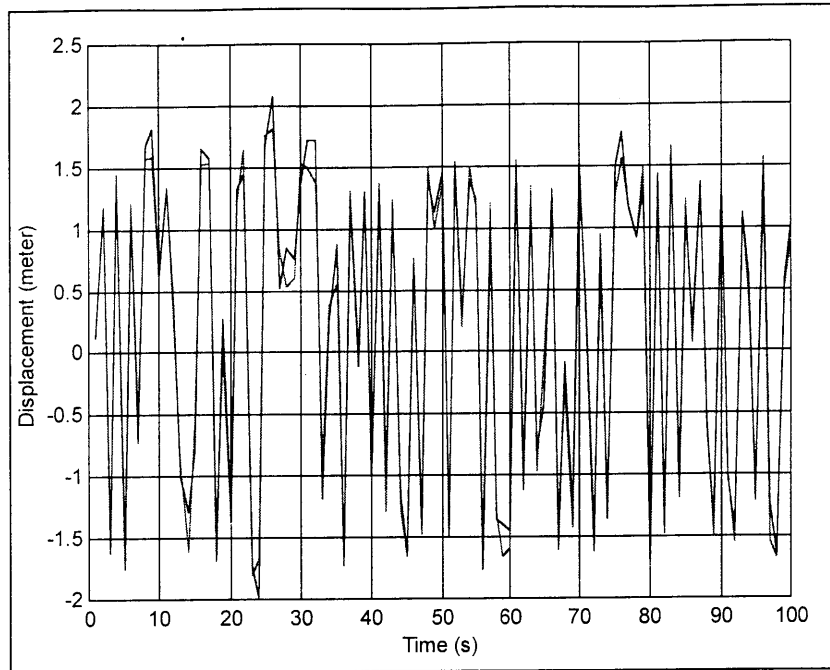


Figure [4.9] Fuzzy model response in comparison with mass-spring system actual output (shows exact modeling).

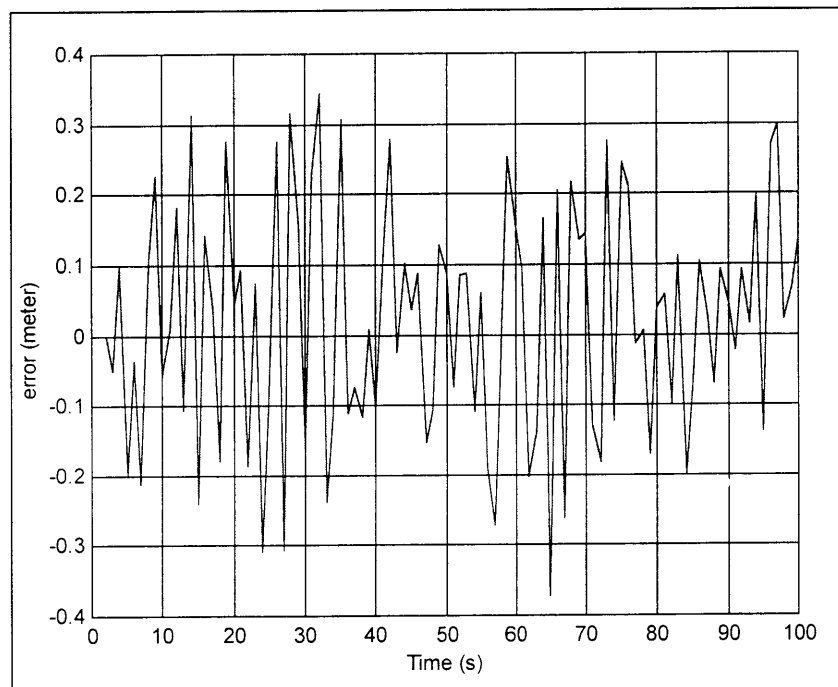


Figure [4.10] Error between fuzzy model response and actual output of the mass-spring system.

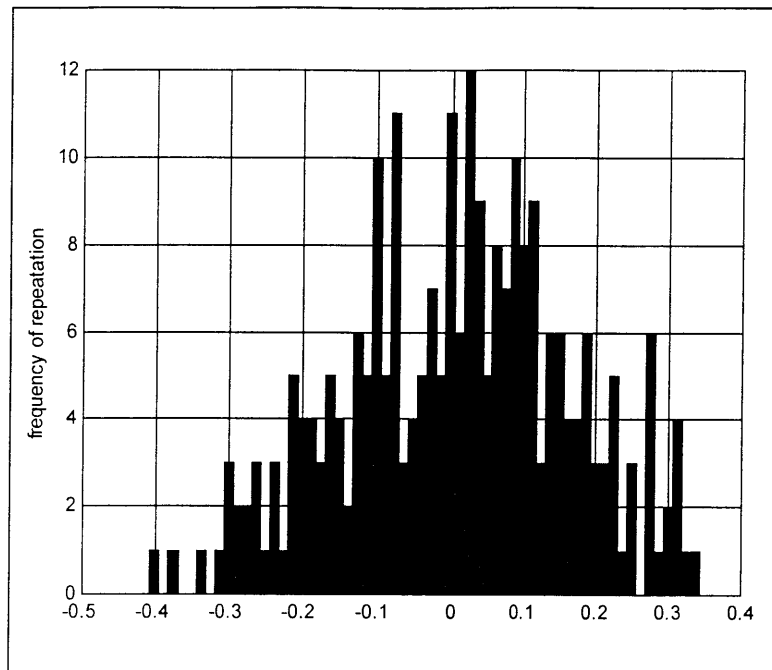


Figure [4.11] Histogram of the error distribution of the mass-spring system.

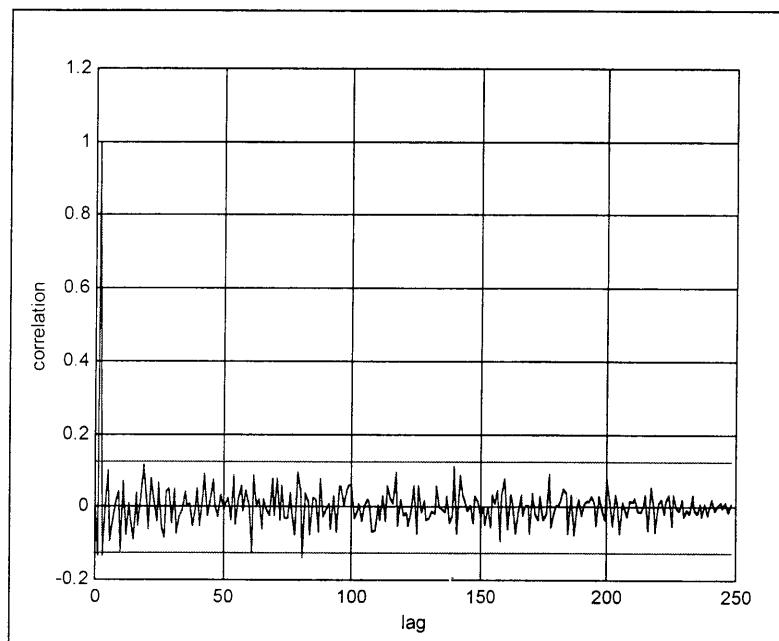


Figure [4.12] Modeling error auto correlation of the mass-spring system.

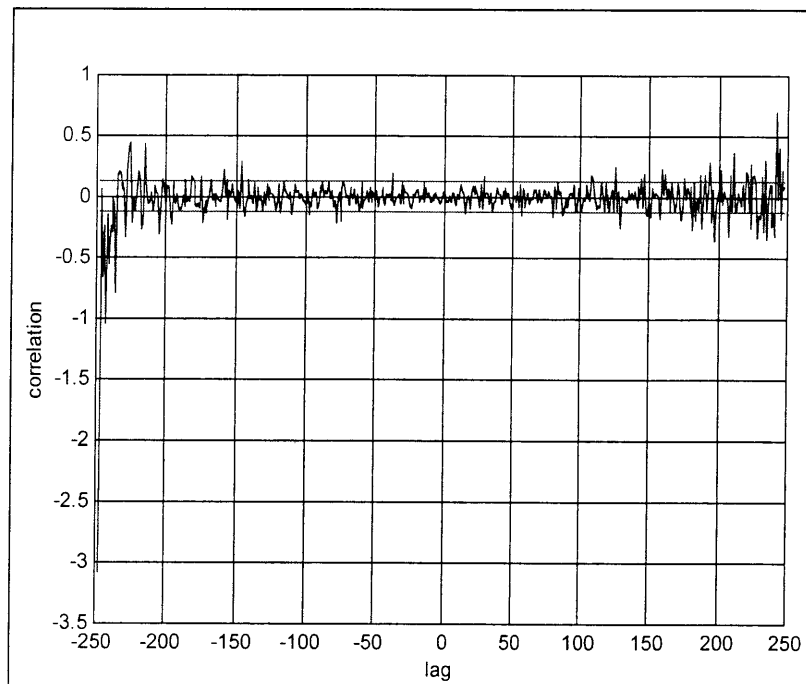


Figure [4.13] Cross-correlation of mass-spring input signal and error.

[B] Antenna System:**Input/Output Training Pattern**

A data set containing 500 samples of data was produced by applying random torques to the different channels, with suitable sampling rate and an amplitude drawn from uniform random distribution in the range $[-1.5, 1.5]$ N/m. These torques are chosen to emulate typical operating conditions. Such block diagram used to produce the identification data was simulated through SIMULINK/MATLAB, using a set of nonlinear differential equations that describes the antenna system as detailed in Appendix [A-2]. Half of the training pattern is used in the modeling of the dynamic system whereas the other half is used to validate the fuzzy models resulted from the modeling. For a typical sequence of torque inputs the response of the antenna is shown in Figure [4.14]

Clustering and Training Pattern

As discussed in section (4.6.6), fuzzy modeling of any dynamical system can be achieved through clustering the training data, where in this simulation example clustering has been applied to the antenna training pattern. Figure [4.15] shows the training pattern to be clustered. The number of clusters is chosen to be three clusters, fuzziness parameter m was kept at 2.2 with a termination criterion $\epsilon = 0.01$. The result of the clustering algorithm is the fuzzy partition matrix and the cluster centers matrix, which will be used to construct the fuzzy model for the antenna system. Figure [4.16] shows the training pattern after applying the clustering algorithm, where it shows clearly the three clusters and their three associated centers.

Neurofuzzy Modeling

Neurofuzzy modeling is applied to the problem of identifying a discrete model of the antenna. A fuzzy model can be constructed from data by using the output of the clustering algorithm and by constructing regressors to form inputs to the neurofuzzy network. Hence a conventional linear difference model with regressors is constructed containing previous inputs and outputs, i.e.

$$\begin{aligned}\varphi(k) &= [T_{\varphi}(k-1), T_{\varphi}(k-2), T_{\psi}(k-1), \varphi(k-1), \varphi(k-2), \psi(k-1), \psi(k-2), \Delta\varphi(k-1)]^T \\ \psi(k) &= [T_{\varphi}(k-1), T_{\psi}(k-1), T_{\psi}(k-2), \varphi(k-1), \varphi(k-2), \psi(k-1), \psi(k-2), \Delta\psi(k-1)]^T\end{aligned}$$

Fuzzy IF-THEN rules can be extracted by projecting the clusters onto the axes and the membership functions of the fuzzy sets generated by point-wise projection of the partition

matrix onto the antecedent variables. Then consequent parameters for each rule are obtained as least squares estimates. When an initial structure is obtained through clustering, then the membership functions and the consequent parameters are tuned to satisfy certain cost function through the learning procedure of the neurofuzzy.

Membership Functions and Associated Fuzzy Rules

As a result, the membership functions of all the inputs (regressors) and output are shown in Figure [4.17] and [4.18] for azimuth angle and elevation angle respectively. The antenna system has six inputs (in terms of its fuzzy model) and two output, hence two groups of seven sets of MFs are shown. Each universe of discourse (set) has three MFs representing the assigned three clusters. Such memberships are representing the range of the inputs.

Fuzzy Model Validation

MEASURE OF ERROR

The output response of the fuzzy model is shown in Figure [4.19] compared to the process output in Figure [4.14], as it is obvious the degree to which the fuzzy model can approximate the process output is reasonable (96% for output (1) and 94% for output (2)). It is obvious that this linear model adequately models the training pattern and it can be more accurate after some adjustment by trial and error techniques.

Figure [4.20] shows the associated error between the two outputs. Further analysis of such error can be shown as well in Figure [4.21], the histogram of error, where it is clear how the error is spread over a quite narrow range.

CORRELATION FUNCTIONS:

Furthermore, we employed correlation tests to check that suitable regressors are uncorrelated with the models residuals. Figure [4.22] shows the cross-correlation of the error signal with the first input signal of the antenna. The correlation in the figure is within the confidence interval, which indicates that the two signals are not correlated.

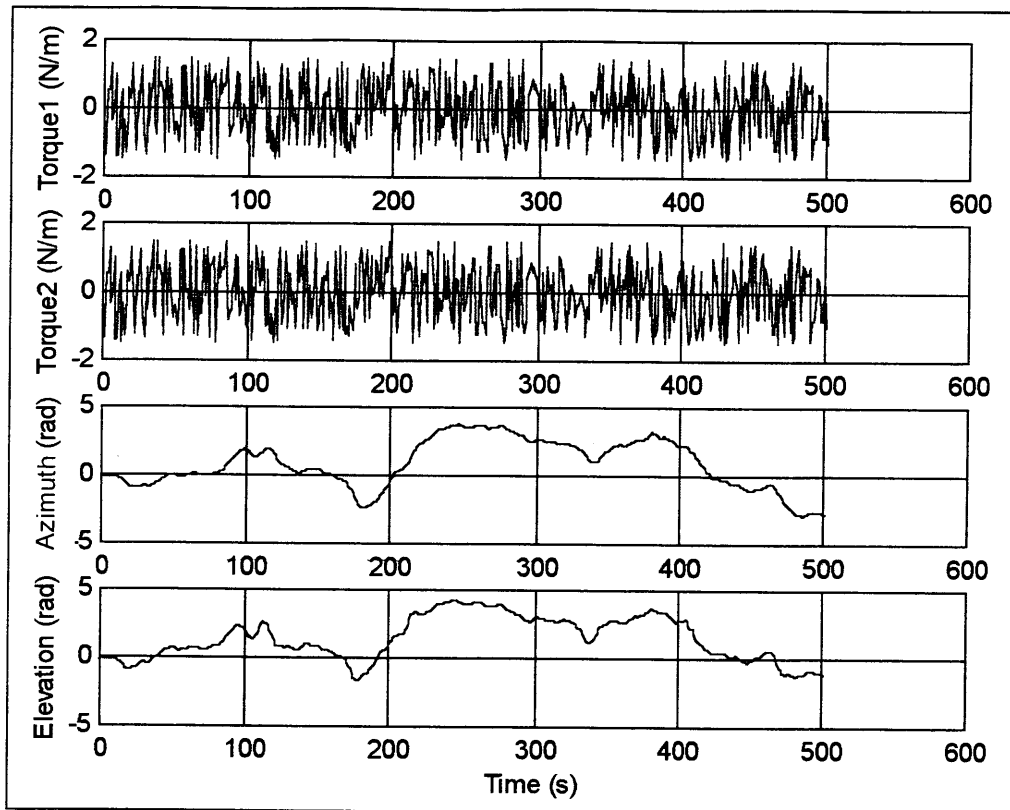


Figure (4.14) Input-Output data pattern used in the identification of an antenna system.

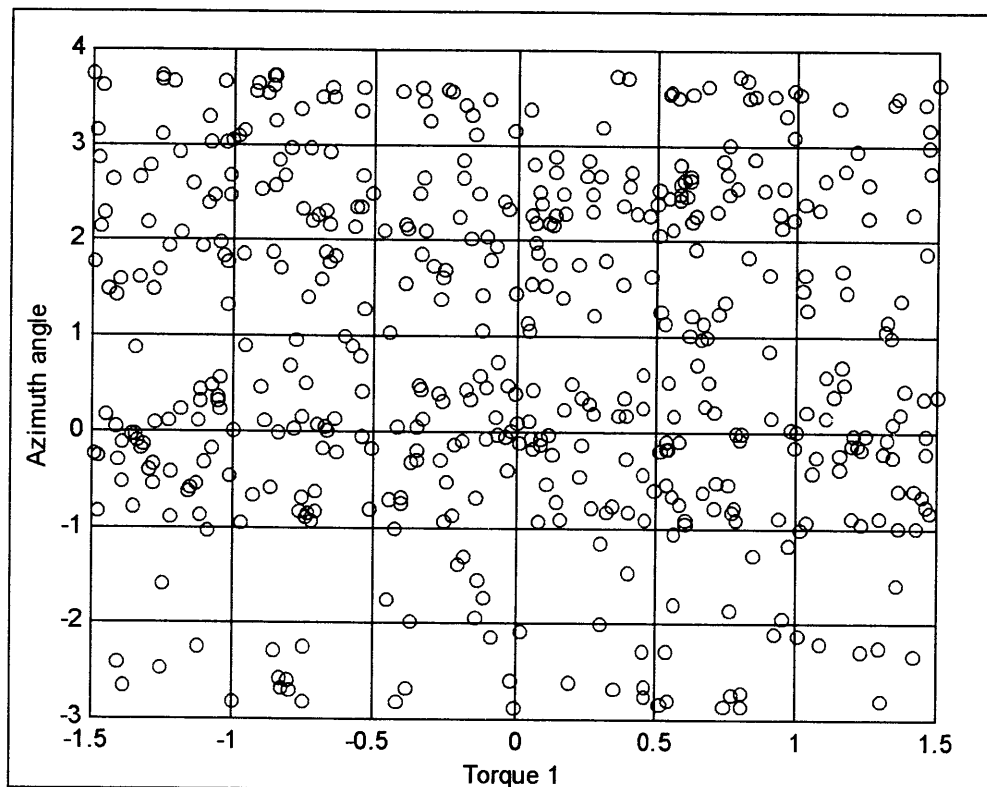


Figure (4.15) Input-output training pattern of the antenna system prior to clustering.

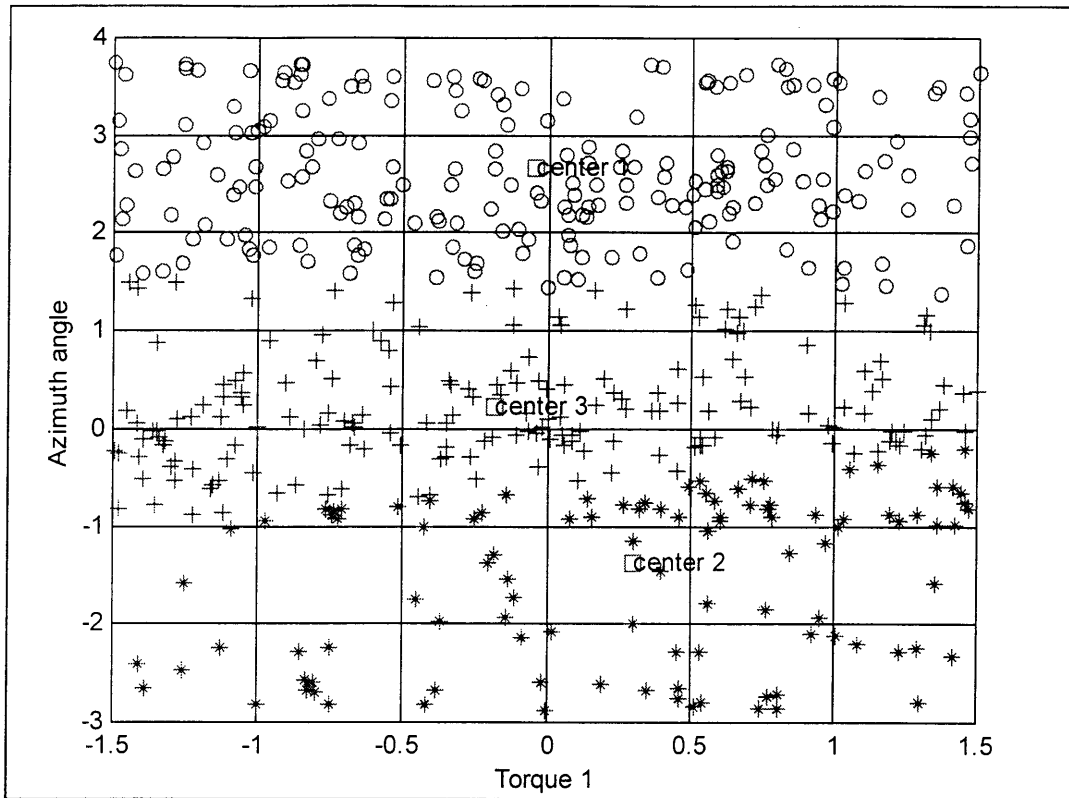


Figure (4.16) Input-Output training pattern of the antenna system after clustering.

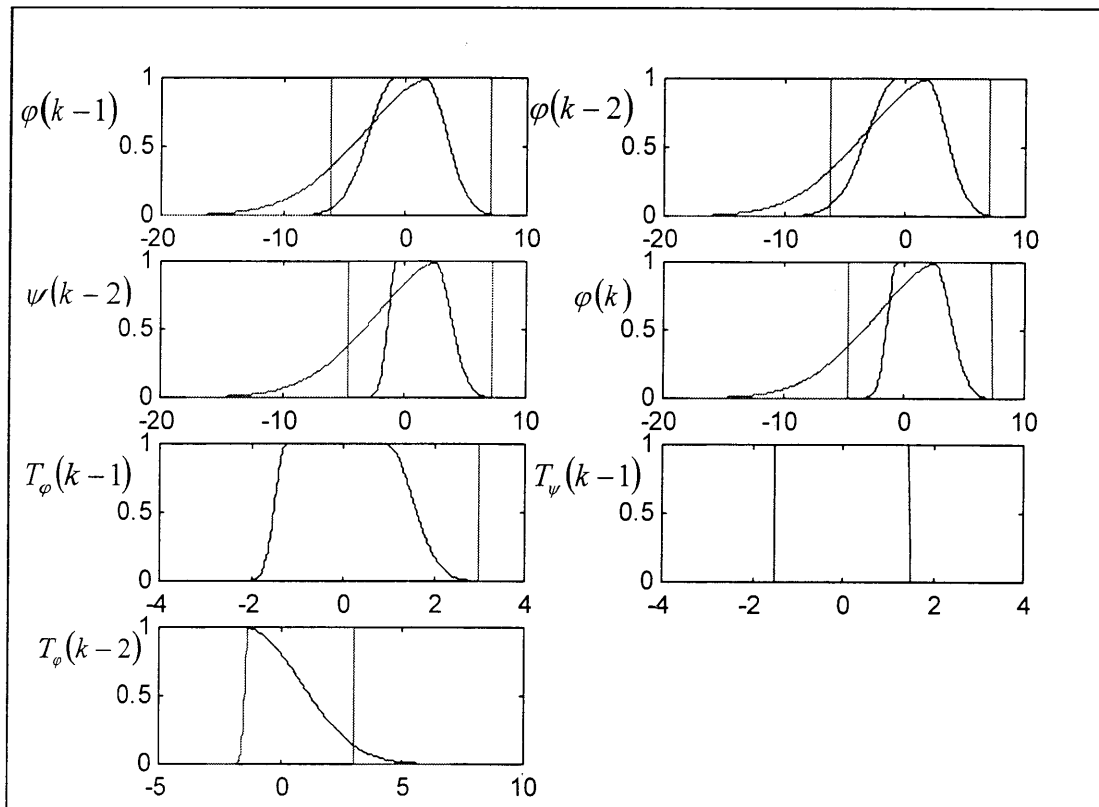


Figure (4.17) Extracted membership functions of all inputs regressors associated with the azimuth angle.

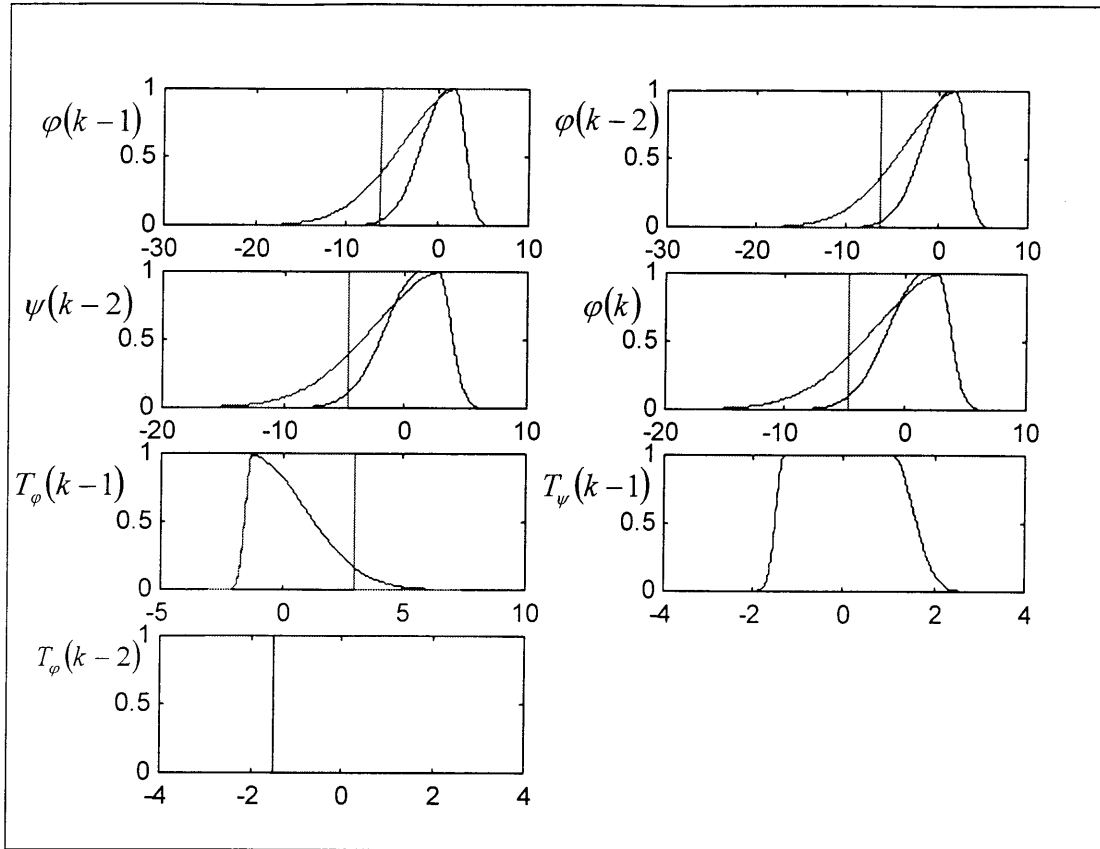


Figure (4.18) Extracted membership functions of all inputs regressors associated with the elevation angle.

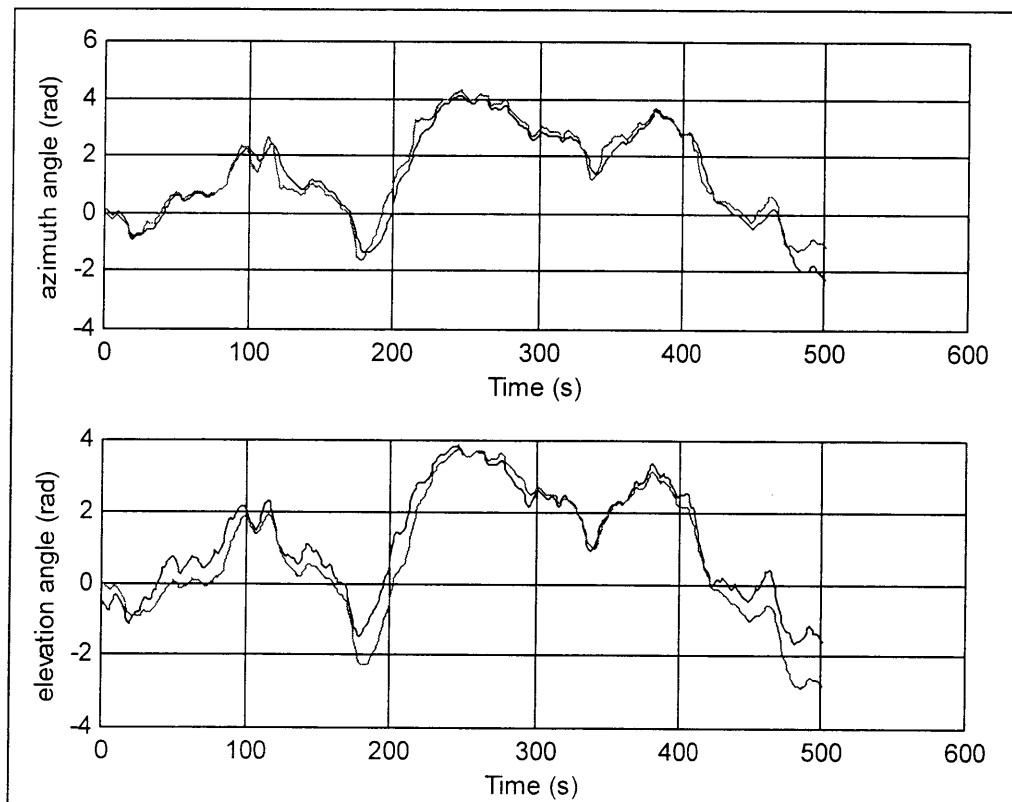


Figure (4.19) Fuzzy model responses (*azimuth and elevation angels*) compared to the antenna outputs.

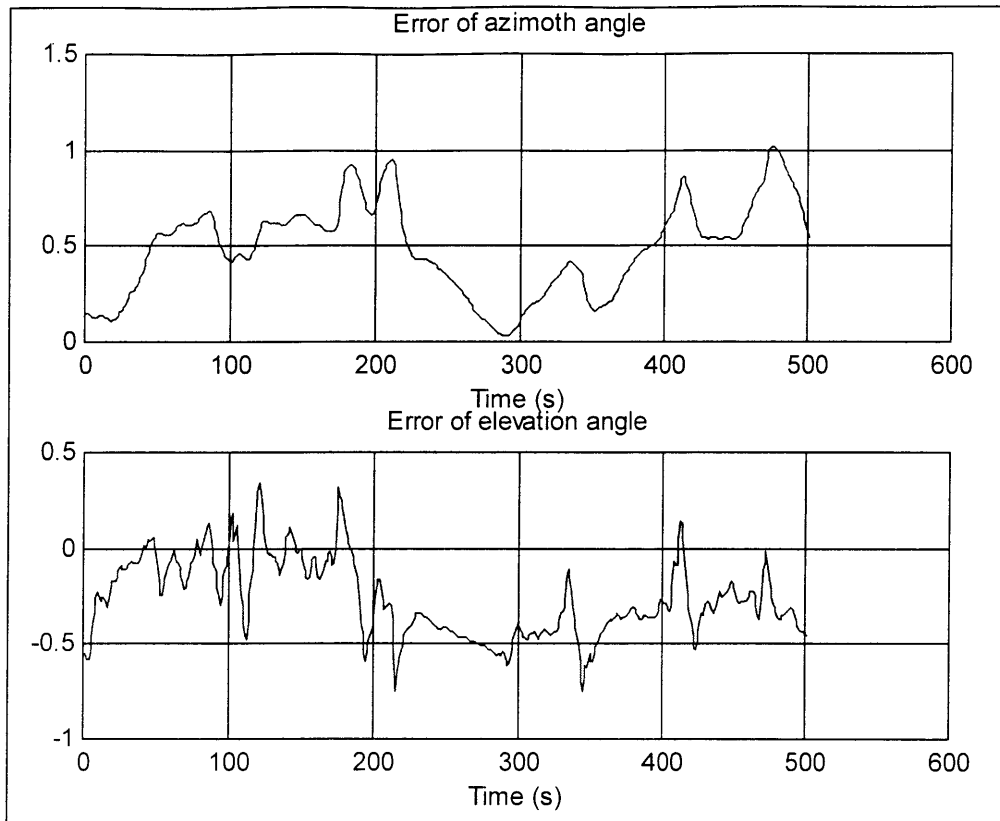


Figure [4.20] Error between the actual output and the fuzzy model output for both azimuth and elevation outputs of the antenna system.

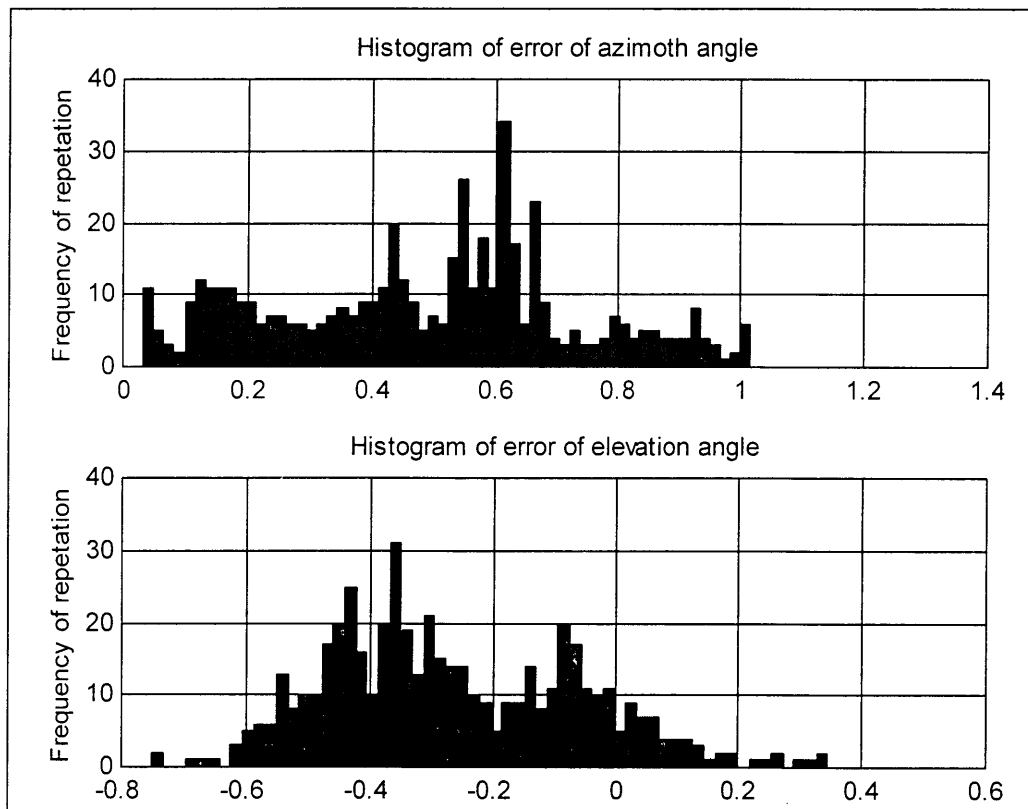


Figure [4.21] Histogram of error distribution of the two errors of the antenna system.

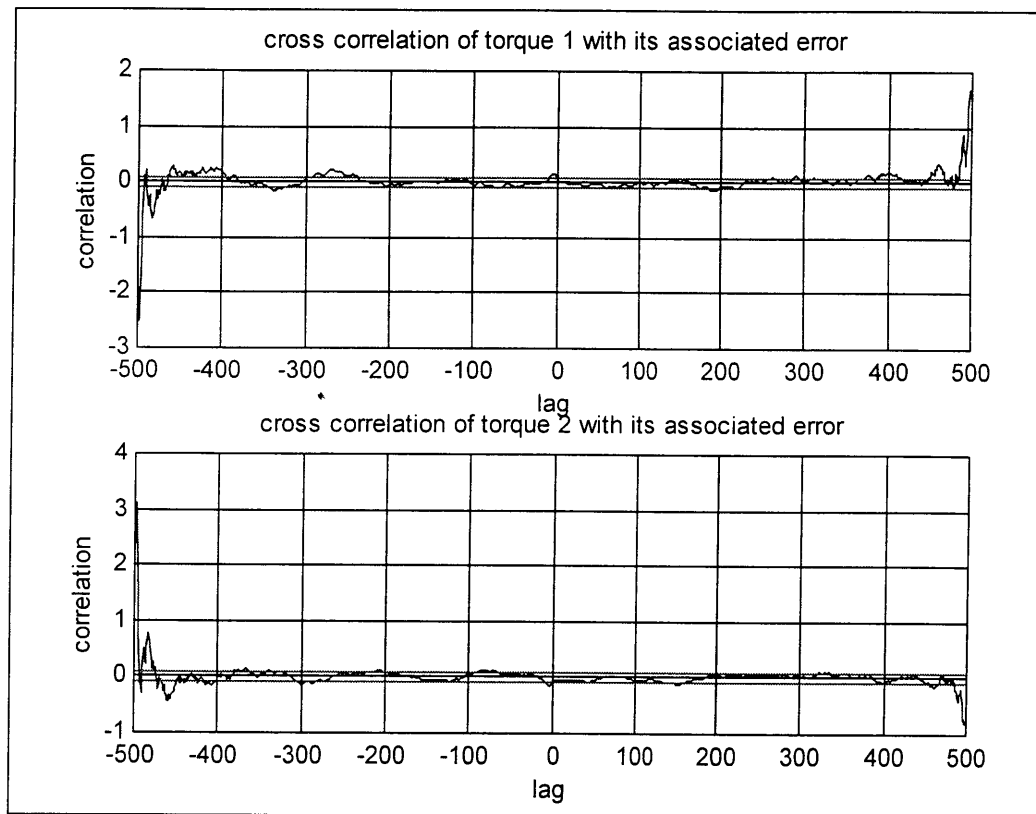


Figure [4.22] Cross correlation of system first input with its associate error of the antenna system.

Fuzzy Sub-models

The presented T-S fuzzy model has been used to identify the nonlinear antenna system. As was mentioned before that the number of rules in the T-S fuzzy model equals the number of clusters in the product space. The consequent of each rule is a local model that approximate the output of the real function for the range of x for which the rule is applicable. As a result of the modeling process, the following rules are obtained for azimuth angle, the same for elevation angle,

Rule 1: IF $T_\varphi(k-1)$ is F_1^1 AND $T_\varphi(k-2)$ is F_2^1 AND $T_\psi(k-1)$ is F_3^1 AND $\varphi(k-1)$ is F_4^1 AND $\varphi(k-2)$ is F_5^1 AND $\psi(k-1)$ is F_6^1 AND $\psi(k-2)$ is F_7^1

$$\text{THEN } \dot{\mathbf{x}}(t) = \mathbf{A}_1 \mathbf{x}(t) + \mathbf{B}_1 \mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}_1 \mathbf{x}(t)$$

Rule 2: IF $T_\varphi(k-1)$ is F_1^2 AND $T_\varphi(k-2)$ is F_2^2 AND $T_\psi(k-1)$ is F_3^2 AND $\varphi(k-1)$ is F_4^2 AND $\varphi(k-2)$ is F_5^2 AND $\psi(k-1)$ is F_6^2 AND $\psi(k-2)$ is F_7^2

$$\text{THEN } \dot{\mathbf{x}}(t) = \mathbf{A}_2 \mathbf{x}(t) + \mathbf{B}_2 \mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}_2 \mathbf{x}(t)$$

Rule 3: IF $T_\varphi(k-1)$ is F_1^3 AND $T_\varphi(k-2)$ is F_2^3 AND $T_\psi(k-1)$ is F_3^3 AND $\varphi(k-1)$ is F_4^3 AND $\varphi(k-2)$ is F_5^3 AND $\psi(k-1)$ is F_6^3 AND $\psi(k-2)$ is F_7^3

$$\text{THEN } \dot{\mathbf{x}}(t) = \mathbf{A}_3 \mathbf{x}(t) + \mathbf{B}_3 \mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}_3 \mathbf{x}(t)$$

where,

$$\mathbf{A}_1 = \begin{bmatrix} 0.09510 & -0.4265 & 0.0300 & -0.2998 & 0.0116 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0.0348 & -0.0273 & 0.8022 & -0.3315 & 0 & 0.0013 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0.9354 & -0.4667 & 0.0525 & -0.0475 & 0.0119 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0.0373 & -0.0384 & 0.0844 & -0.3715 & 0 & -0.0002 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 0.9020 & -0.4099 & 0.0142 & -0.0157 & 0.0108 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0.1294 & -0.1098 & 0.7887 & -0.2987 & 0 & 0.0002 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} 0.0128 & 0 \\ 0 & 0 \\ 0 & 0.0240 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0.0128 & 0 \\ 0 & 0 \\ 0 & 0.0137 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B}_3 = \begin{bmatrix} 0.0118 & 0 \\ 0 & 0 \\ 0 & 0.0028 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{D}_1 = \mathbf{D}_2 = \mathbf{D}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where the $\mathbf{C}$ and $\mathbf{D}$ matrix are common for all of the three fuzzy sub-models, and the $\mathbf{D}$ matrix is equal to zero. Also, the elevation angle output is of the same above structure.

4.11 Conclusions

Chapter four has been concentrated on the modeling of nonlinear dynamic systems via the utilization of the well known fuzzy modeling paradigm, the Takagi-Sugeno T-S modeling technique.

The central objective of this chapter has been the modeling technique via T-S modeling, for the purpose of H^∞ robust control system later design of two nonlinear systems, a SISO mass-spring system and a multivariable antenna system. T-S models depends heavily on some initial membership centers of the universe of discourse of the used fuzzy variables, such centers have been obtained through the used of clustering algorithm. Once such centers are known, the fuzzy system can establish initial membership centers through which they are updated via some learning mechanism.

One of the advantages of T-S modeling is that systems can be modeled by a few rules, and consequently few linear sub-models. This advantage has overcome the problem of the large number of rules in the fuzzy modeling.

Fuzzy models have also been verified and validated through some standard validation techniques. Shows clearly the successful ability of T-S modeling to model nonlinear systems with a good degree of accuracy.

Chapter Five:

Fuzzy H^∞ Robust Controller Synthesis for Nonlinear Systems

5.1 Introduction

In the past few years, considerable attention has been paid to the problems of robust stabilization and robust performance of dynamic systems described by fuzzy models. Hence, one of the most important requirements for a fuzzy control system is the robustness property.

An attempt has been made to create a bridge between two important control design techniques, *Robust* and *Fuzzy Control Design*. This will help to supply H^∞ control design with more intelligence and to supply fuzzy control design with better performance (i.e. robustness). The bridge is achieved by first introducing a fuzzy linear T-S model to approximate a nonlinear system. Hence, a neurofuzzy controller is introduced to stabilize the nonlinear system, and at the same time to eliminate the effect of external disturbance below a prescribed level. In this way a desired H^∞ control performance can be guaranteed. Such design of robust fuzzy control is solved via Linear Matrix Inequality (LMI) techniques. At the end, a simple but practical algorithm is obtained for robust performance control design of nonlinear system.

We shall address the problem of robust H^∞ disturbance attenuation for a class of nonlinear system modeled via T-S fuzzy models in which both robust stability and a prescribed H^∞ performance are to be achieved. Furthermore, the robust stabilization problem and robust H^∞ control problem will be directly addressed using LMI approach.

In this context, the chapter has been divided into four main sections. Section one is the chapter introduction, where as section two presents the basis of robust fuzzy control through the problem formulation of the closed loop Lyapunov function, here the formulation of the robust H^∞ control is represented as an LMI problem. Section three presents the design of H^∞ robust control for the nonlinear mass-spring system and the nonlinear antenna system which have been modeled via T-S fuzzy modeling in chapter four. Finally section four draws few points of conclusion.

5.2 Robust Fuzzy Control

A general multivariable nonlinear dynamic system is represented by the following form,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) + \mathbf{w}(t) \quad (5.2)$$

in which

$$\mathbf{x}(t) = [x_1(t) \ x_2(t) \dots x_n(t)]^T \in \mathcal{R}^{n \times 1} \quad (5.3)$$

presents the system state vector,

$$\mathbf{u}(t) = [u_1(t) \ u_2(t) \dots u_m(t)]^T \in \mathcal{R}^{m \times 1} \quad (5.4)$$

denotes the control input vector and,

$$\mathbf{w}(t) = [w_1(t) \ w_2(t) \dots w_n(t)]^T \in \mathcal{R}^{n \times 1} \quad (5.5)$$

denotes the unknown disturbances vector acting at the system output and being characterized with a known upper bound $w_{ub} \geq \|w(t)\|$, and $f(x)$ and $g(x)$ depend on the system state $\mathbf{x}(t)$.

A fuzzy dynamic model has been proposed by Takagi and Sugeno in their well known paper [9], and used to represent local linear input-output relations of the nonlinear system under consideration.

5.2.1 Takagi-Sugeno Fuzzy Model Formulation

As it was mentioned earlier in chapter four, we represent the nonlinear system by a T-S fuzzy model that is described by IF-THEN fuzzy rules. Hence locally representing linear input-output relations of the system. In this sense, the i th rule of this fuzzy model for the nonlinear system described by Equ. [5.2] is of the following form,

Dynamic Part:

$$\begin{aligned} \text{Rule } i: \quad & \text{IF } z_1(t) \text{ is } M_{i1} \dots \text{AND } z_g(t) \text{ is } M_{ig} \quad i = 1, 2, \dots, r \\ & \text{THEN } \dot{\mathbf{x}}(t) = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t) + \mathbf{w}(t), \end{aligned} \quad (5.6)$$

Output Part:

$$\begin{aligned} \text{Rule } i: \quad & \text{IF } z_1(t) \text{ is } M_{i1} \dots \text{AND } z_g(t) \text{ is } M_{ig} \quad i = 1, 2, \dots, r \\ & \text{THEN } \mathbf{y}(t) = \mathbf{C}_i \mathbf{x}(t) \end{aligned} \quad (5.7)$$

where,

$$\mathbf{z}(t) = [z_1(t), \ z_2(t), \dots, z_n(t)]^T \in \mathcal{R}^{n \times 1} \quad (5.8)$$

The vectors $\mathbf{x}(t)$, $\mathbf{u}(t)$, and $\mathbf{y}(t)$ denote the state, input and output vectors respectively, and $\mathbf{A}_i \in \mathcal{R}^{n \times n}$, $\mathbf{B}_i \in \mathcal{R}^{n \times m}$, $\mathbf{C}_i \in \mathcal{R}^{o \times n}$. Each variable $z_i(t)$ is a known parameter and may be function of state variables, external disturbances, and/or time. In Equis. [5.6] r is the number

of IF-THEN rules, M_{ig} is the fuzzy membership function associated with the i^{th} rule and g^{th} parameter component.

There are two functions of $\mathbf{z}(t)$ associated with each rule. The first function is called the truth value (or the degree of fulfillment or the weight), where the truth value for an i^{th} rule is defined by,

$$\mu_i(\mathbf{z}(t)) = \prod_{j=1}^g M_{ij}(z_j(t)) \quad (5.9)$$

where $M_{ij}(z_j(t))$ is the grade of membership of $z_j(t)$. The possibility that the i^{th} rule will fire is given by the product of all the membership functions associated with the i^{th} rule. Each μ_i is a non-negative function and that the truth value of at least one rule is always nonzero. The second function is called the firing probability. The firing probability for an i^{th} rule is hence, defined by,

$$h_i(\mathbf{z}(t)) = \frac{\mu_i(\mathbf{z}(t))}{\sum_{i=1}^r \mu_i(\mathbf{z}(t))} \quad (5.10)$$

which is always a well defined function taking values between 0 and 1, and the sum of all the firing probabilities is identically equal to unity.

In Equ. [5.9], it is assumed that,

$$\mu_i(\mathbf{z}(t)) \geq 0 \quad (5.11)$$

$$\text{and } \sum_{i=1}^r \mu_i(\mathbf{z}(t)) > 0 \quad (5.12)$$

for all t . Therefore, we get,

$$h_i(\mathbf{z}(t)) \geq 0 \quad (5.13)$$

$$\text{and } \sum_{i=1}^r h_i(\mathbf{z}(t)) = 1 \quad (5.14)$$

For a given a pair of $(\mathbf{x}(t), \mathbf{u}(t))$, using the center of gravity method already presented in chapter three for defuzzification, the final output of the T-S fuzzy system is then inferred as follows,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \frac{\sum_{i=1}^r \mu_i(\mathbf{z}(t)) [\mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t)]}{\sum_{i=1}^r \mu_i(\mathbf{z}(t))} + \mathbf{w}(t) \\ &= \sum_{i=1}^r h_i(\mathbf{z}(t)) [\mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t)] + \mathbf{w}(t)\end{aligned}\quad (5.15)$$

and

$$\mathbf{y} = \sum_{i=1}^r h_i(\mathbf{z}(t)) \mathbf{C}_i \mathbf{x} \quad (5.16)$$

5.2.2 Basic Stability Condition

Stability is the most important issue in control systems. Obviously, it is important to specify control parameters K_j in the fuzzy controller such that the stability for the closed-loop nonlinear system can be guaranteed. For the fuzzy open loop systems of (Equ. [5.15]) are defined as follows:

Once $\mathbf{u}(t) = 0$ then,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \frac{\sum_{i=1}^r \mu_i(\mathbf{z}(t)) [\mathbf{A}_i \mathbf{x}(t)]}{\sum_{i=1}^r \mu_i(\mathbf{z}(t))} \\ &= \sum_{i=1}^r h_i(\mathbf{z}(t)) [\mathbf{A}_i \mathbf{x}(t)]\end{aligned}\quad (5.17)$$

Each linear consequent equation represented by $\mathbf{A}_i \mathbf{x}(t)$ is called “sub-system”. Stability condition for ensuring the stability of the system in Equ. [5.17] is found [11]. Hence, the sub-systems are asymptotically stable in the large if there exists a common positive definite matrix $\mathbf{P}$ such that,

$$\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i < 0 \quad (5.18)$$

for $i=1,2,\dots, r$, that is, for all the sub-systems defined by the T-S fuzzy models.

Equ. [5.18] depends only on the system sub-models (i.e. $\mathbf{A}_i$). In other words, it does not depend on the disturbance $\mathbf{w}(t)$. This is reduced to the Lyapunov stability theorem for linear systems once $r=1$. That gives a sufficient condition for ensuring the stability of the system defined by Equ. [5.17].

The problem of finding a common Lyapunov function $\mathbf{P}$ in Equ. [5.18] can be solved numerically by convex programming algorithms involving LMI's. These algorithms are extremely efficient in practice, especially once the system under control is of a large number of states and of a multivariable nature.

Therefore from Equ. [5.2] and Equ. [5.15] we get,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t) + \mathbf{w}(t) \\ &= \sum_{i=1}^r h_i(\mathbf{z}(t))[\mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t)] \\ &+ \left\{ \left(f(\mathbf{x}) - \sum_{i=1}^r h_i(\mathbf{z}(t))[\mathbf{A}_i \mathbf{x}(t)] \right) + \left(g(\mathbf{x}) - \sum_{i=1}^r h_i(\mathbf{z}(t))(\mathbf{B}_i) \right) \mathbf{u}(t) \right\} + \mathbf{w}(t)\end{aligned}\quad (5.19)$$

where,

$$\left\{ \left(f(\mathbf{x}) - \sum_{i=1}^r h_i(\mathbf{z}(t))[\mathbf{A}_i \mathbf{x}(t)] \right) + \left(g(\mathbf{x}) - \sum_{i=1}^r h_i(\mathbf{z}(t))(\mathbf{B}_i) \right) \mathbf{u}(t) \right\} \quad (5.20)$$

denotes the approximation error between the nonlinear system Equ. [5.2] and the fuzzy model Equ. [5.15].

5.2.3 Fuzzy Controller

A typical fuzzy controller of $\mathbf{u}=\mathbf{K}\mathbf{x}$ is employed to deal with the control system design. In terms of fuzzy rules this is given by,

Control Rule j:

$$\begin{aligned}\text{Rule } j: \quad &\text{IF } z_l(t) \text{ is } M_{jl} \dots \text{ AND } z_g(t) \text{ is } M_{jg} \\ &\text{THEN } \mathbf{u}(t) = \mathbf{K}_j \mathbf{x}(t)\end{aligned}\quad (5.21)$$

For $j=1,2,\dots,r$, the overall fuzzy controller is defined by,

$$\mathbf{u}(t) = \frac{\sum_{j=1}^r \mu_j(\mathbf{z}(t))[\mathbf{K}_j \mathbf{x}(t)]}{\sum_{j=1}^r \mu_j(\mathbf{z}(t))} \quad (5.22)$$

$$= \sum_{j=1}^r h_j(\mathbf{z}(t))[\mathbf{K}_j \mathbf{x}(t)] \quad (5.23)$$

where $h_j(\mathbf{z}(t))$ was defined by Equ. [5.10] and $\mathbf{K}_j$ are the control parameters for j from 1 up to r .

5.2.4 Closed Loop Dynamics of Fuzzy Systems

In order to obtain the entire closed loop dynamics of the nonlinear system, Equ. [5.23] is substituted into Equ. [5.19]. This yields the closed-loop nonlinear control system given below by,

$$\begin{aligned}
 \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t) + \mathbf{w}(t) \\
 &= \sum_{i=1}^r \sum_{j=1}^r h_i(\mathbf{z}(t))h_j(\mathbf{z}(t))(\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_j)\mathbf{x}(t) \\
 &\quad + \left(f(\mathbf{x}) - \sum_{i=1}^r h_i(\mathbf{z}(t))\mathbf{A}_i\mathbf{x}(t) \right) \\
 &\quad + \sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t))(\mathbf{g}(\mathbf{x}) - \mathbf{B}_i)\mathbf{K}_j\mathbf{x}(t) + \mathbf{w}(t) \\
 &= \sum_{i=1}^r \sum_{j=1}^r h_i(\mathbf{z}(t))h_j(\mathbf{z}(t))(\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_j)\mathbf{x}(t) + \Delta f + \Delta g + \mathbf{w}(t)
 \end{aligned} \tag{5.24}$$

where, Δf and Δg are the error between the fuzzy model and the actual system.

$$\Delta f = \left(f(\mathbf{x}) - \sum_{i=1}^r h_i(\mathbf{z}(t))\mathbf{A}_i\mathbf{x}(t) \right) \tag{5.25}$$

and

$$\Delta g = \sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t))(\mathbf{g}(\mathbf{x}) - \mathbf{B}_i)\mathbf{K}_j\mathbf{x}(t) \tag{5.26}$$

If there exist bounding matrices $\Delta\mathbf{A}_i$ and $\Delta\mathbf{B}_i$ in such away that,

$$\|\Delta f\| \leq \left\| \sum_{i=1}^r h_i(\mathbf{z}(t))\Delta\mathbf{A}_i\mathbf{x}(t) \right\| \tag{5.27}$$

and

$$\|\Delta g\| \leq \left\| \sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t))\Delta\mathbf{B}_i\mathbf{K}_j\mathbf{x}(t) \right\| \tag{5.28}$$

for all trajectory of $\mathbf{x}(t)$ and the bounding matrices $\Delta\mathbf{A}_i$ and $\Delta\mathbf{B}_i$ are described by,

$$\begin{bmatrix} \Delta \mathbf{A}_i \\ \Delta \mathbf{B}_i \end{bmatrix} = \begin{bmatrix} \delta_i \mathbf{A}_p \\ \eta_i \mathbf{B}_p \end{bmatrix} \quad (5.29)$$

where $\|\delta_i\| \leq 1$ and $\|\eta_i\| \leq 1$, for $i = 1, 2, \dots, r$

Now, if we assume $g = n$ and

$$z_1(t) = x_1(t), \quad z_2(t) = x_2(t) \quad \dots, z_n(t) = x_n(t) \quad (5.30)$$

then the plant rule can be represented as

Dynamic Part:

$$\begin{aligned} \text{Rule } i: \quad & \text{IF } x_1(t) \text{ is } M_{i1} \dots \text{ AND } x_n(t) \text{ is } M_{in} \\ & \text{THEN } \dot{\mathbf{x}}(t) = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t) + \mathbf{w}(t), \end{aligned} \quad (5.31)$$

In reference to the above assumptions, Equ. [5.27] and Equ. [5.28], are the worst case representations for the approximation error if there exist $\Delta \mathbf{A}_i (= \delta_i \mathbf{A}_p)$ and $\Delta \mathbf{B}_i (= \eta_i \mathbf{B}_p)$ such that Equ. [5.27] & Equ. [5.28] hold for some $\|\delta_i\| \leq 1$ and $\|\eta_i\| \leq 1$ (for $i=1, 2, \dots, r$), where $\mathbf{A}_p$ and $\mathbf{B}_p$ are bounding matrices.

While the above assumption in consideration, we get,

$$\begin{aligned} (\Delta f)^T (\Delta f) &= \left(f(\mathbf{x}) - \sum_{i=1}^r h_i(\mathbf{z}(t)) \mathbf{A}_i \mathbf{x}(t) \right)^T \left(f(\mathbf{x}) - \sum_{i=1}^r h_i(\mathbf{z}(t)) \mathbf{A}_i \mathbf{x}(t) \right) \\ &\leq \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \Delta \mathbf{A}_i \mathbf{x}(t) \right)^T \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \Delta \mathbf{A}_i \mathbf{x}(t) \right) \\ &= \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \delta_i \mathbf{A}_p \mathbf{x}(t) \right)^T \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \delta_i \mathbf{A}_p \mathbf{x}(t) \right) \\ &\leq (\mathbf{A}_p \mathbf{x}(t))^T (\mathbf{A}_p \mathbf{x}(t)) \end{aligned} \quad (5.32)$$

and

$$\begin{aligned} & (\Delta g)^T (\Delta g) \\ &= \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) (g(\mathbf{x}) - \mathbf{B}_i) \mathbf{K}_j \mathbf{x}(t) \right)^T \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) (g(\mathbf{x}) - \mathbf{B}_i) \mathbf{K}_j \mathbf{x}(t) \right) \end{aligned}$$

$$\begin{aligned}
&\leq \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) (g(\mathbf{x}) - \Delta \mathbf{B}_i) \mathbf{K}_j \mathbf{x}(t) \right)^T \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) (g(\mathbf{x}) - \Delta \mathbf{B}_i) \mathbf{K}_j \mathbf{x}(t) \right) \\
&= \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) (g(\mathbf{x}) - \eta_i \mathbf{B}_p) \mathbf{K}_j \mathbf{x}(t) \right)^T \left(\sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) (g(\mathbf{x}) - \eta_i \mathbf{B}_p) \mathbf{K}_j \mathbf{x}(t) \right) \\
&\leq \left(\sum_{j=1}^r h_j(\mathbf{z}(t)) \mathbf{B}_p \mathbf{K}_j \mathbf{x}(t) \right)^T \left(\sum_{j=1}^r h_j(\mathbf{z}(t)) \mathbf{B}_p \mathbf{K}_j \mathbf{x}(t) \right) \quad (5.33)
\end{aligned}$$

i.e., the approximation error in the closed-loop nonlinear system is bounded by the specified structured bounding matrices $\mathbf{A}_p$ and $\mathbf{B}_p$.

5.2.5 H^∞ Control

Disturbance effect is an important issue in the design of robust control. In this thesis, we assume that $\mathbf{w}(t)$ is an unknown but bounded disturbance. However, the effect of $\mathbf{w}(t)$ will deteriorate the control performance of fuzzy control system. Therefore, how to eliminate the effect of $\mathbf{w}(t)$ to guarantee the control performance is the main objective.

Since H^∞ control is the most important control design to efficiently eliminate the effect of $\mathbf{w}(t)$ on the control system, it will be employed to deal with the robust performance control.

Considering the following H^∞ control performance,

$$\frac{\int_0^{t_f} \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) dt}{\int_0^{t_f} \mathbf{w}^T(t) \mathbf{w}(t) dt} < \gamma^2 \quad (5.34)$$

or,

$$\int_0^{t_f} \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) dt < \gamma^2 \int_0^{t_f} \mathbf{w}^T(t) \mathbf{w}(t) dt \quad (5.35)$$

where t_f denotes the terminal time of the control, γ is a prescribed value which denotes the worst effect of $\mathbf{w}(t)$ on $\mathbf{x}(t)$, and $\mathbf{Q}$ is a positive-definite weighting matrix. The physical meaning of Equ. [5.35] is that the effect of the disturbance $\mathbf{w}(t)$ must be attenuated below a desired level γ from the view point of energy, no matter what $\mathbf{w}(t)$ is. The desired level of γ is chosen as a positive small value less than 1 for attenuation of $\mathbf{w}(t)$.

The inequality given by Equ. [5.35] can be seen as bounded disturbance and bounded state but with a prescribed gain γ . If the initial condition is also considered, the inequality Equ. [5.35] is modified as,

$$\int_0^T \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) dt < \mathbf{x}^T(0) \mathbf{P} \mathbf{x}(0) + \gamma^2 \int_0^T \mathbf{w}^T(t) \mathbf{w}(t) dt \quad (5.36)$$

where $\mathbf{P}$ is some symmetric positive-definite weighting function.

From the above analysis, the design purpose of the fuzzy control system is to specify a linear fuzzy controller such that both the stability of fuzzy linear control and the H^∞ control performance in Equ. [5.36] with a prescribed attenuation level γ are guaranteed.

The robustness optimization is to achieve a minimum γ^2 in Equ. [5.36] to obtain maximum elimination of the effect of the disturbance $\mathbf{w}(t)$. For the nonlinear system Equ. [5.2], this design problem is reduced to specify a stabilizable fuzzy control $\mathbf{K}$ as will be discussed in the coming section.

5.2.6 H^∞ Control Design via Fuzzy Linear Control

From the description in the above section, the design purpose is how to specify a fuzzy linear control law in Equ. [5.23] for the nonlinear system in Equ. [5.24] with the guaranteed H^∞ control performance in Equ. [5.36].

Since the system in Equ. [5.24] is nonlinear system, then choosing a LYAPUNOV function of the form,

$$V(t) = \mathbf{x}^T(t) \mathbf{P} \mathbf{x}(t) \quad (5.37)$$

where the weighting matrix $\mathbf{P} = \mathbf{P}^T > 0$.

The time derivative of $V(t)$ is

$$\dot{V}(t) = \dot{\mathbf{x}}^T(t) \mathbf{P} \mathbf{x}(t) + \mathbf{x}^T(t) \mathbf{P} \dot{\mathbf{x}}(t) \quad (5.38)$$

For the $\dot{\mathbf{x}}(t)$ in Equ. [5.38], and by substituting Equ. [5.24] for it, $\dot{V}(t)$ is expressed as:

$$\begin{aligned}
\dot{V}(t) &= \sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) \\
&\cdot \left\{ \mathbf{x}^T(t) \left(\mathbf{P}(\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_j) + (\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_j)^T \mathbf{P} \right) \mathbf{x}(t) \right\} + (\Delta f)^T \mathbf{P} \mathbf{x}(t) + \mathbf{x}^T(t) \mathbf{P} (\Delta f) + (\Delta g)^T \mathbf{P} \mathbf{x}(t) \\
&+ \mathbf{x}^T(t) \mathbf{P} (\Delta g) + \mathbf{x}^T(t) \mathbf{P} \mathbf{w}(t) + \mathbf{w}^T(t) \mathbf{P} \mathbf{x}(t) \\
&\leq \sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) \\
&\cdot \left\{ \mathbf{x}^T(t) \left(\mathbf{P}(\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_j) + (\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_j)^T \mathbf{P} \right) \mathbf{x}(t) + (\Delta f)^T (\Delta f) + \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) + (\Delta g)^T (\Delta g) \right. \\
&+ \left. \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) \right\} + \mathbf{x}^T(t) \mathbf{P} \mathbf{w}(t) + \mathbf{w}^T(t) \mathbf{P} \mathbf{x}(t) \\
&\leq \sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) \\
&\cdot \left\{ \mathbf{x}^T(t) \left(\mathbf{P}(\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_j) + (\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_j)^T \mathbf{P} \right) \mathbf{x}(t) + (\mathbf{A}_p \mathbf{x}(t))^T (\mathbf{A}_p \mathbf{x}(t)) + (\mathbf{B}_p \mathbf{K}_j \mathbf{x}(t))^T (\mathbf{B}_p \mathbf{K}_j \mathbf{x}(t)) \right. \\
&+ \left. 2 \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) \right\} + \mathbf{x}^T(t) \mathbf{P} \mathbf{w}(t) + \mathbf{w}^T(t) \mathbf{P} \mathbf{x}(t) \\
&\leq \sum_{i=1}^r h_i(\mathbf{z}(t)) \sum_{j=1}^r h_j(\mathbf{z}(t)) \\
&\cdot \left\{ \mathbf{x}^T(t) \left(\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i + \mathbf{P} \mathbf{B}_i \mathbf{K}_j + \mathbf{K}_j^T \mathbf{B}_i^T \mathbf{P} + \mathbf{A}_p^T \mathbf{A}_p + (\mathbf{B}_p \mathbf{K}_j)^T (\mathbf{B}_p \mathbf{K}_j) + 2 \mathbf{P} \mathbf{P} \right) \mathbf{x}(t) \right\} \\
&+ \mathbf{x}^T(t) \mathbf{P} \mathbf{w}(t) + \mathbf{w}^T(t) \mathbf{P} \mathbf{x}(t) \tag{5.39}
\end{aligned}$$

Hence, if the fuzzy controller Equ. [5.23] is employed in the nonlinear system Equ. [5.2] and there exists a positive-definite matrix $\mathbf{P} = \mathbf{P}^T > 0$ such that the following matrix inequalities:

$$\begin{aligned}
&\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i + \mathbf{P} \mathbf{B}_i \mathbf{K}_j + \mathbf{K}_j^T \mathbf{B}_i^T \mathbf{P} + \mathbf{A}_p^T \mathbf{A}_p \\
&\quad + (\mathbf{B}_p \mathbf{K}_j)^T (\mathbf{B}_p \mathbf{K}_j) + \left(2 + \frac{1}{\gamma^2} \right) \mathbf{P} \mathbf{P} + \mathbf{Q} < 0 \tag{5.41}
\end{aligned}$$

are satisfied for $i, j=1, 2, \dots, r$ then the closed-loop nonlinear system Equ. [5.24] is bounded and the H^∞ control performance of Equ. [5.36] is guaranteed for a prescribed γ^2 .

From Equ. [5.40], we get,

$$\begin{aligned} \dot{V}(t) &\leq \sum_{i=1}^r \sum_{j=1}^r h_i(\mathbf{z}(t)) h_j(\mathbf{z}(t)) \\ &\left\{ \mathbf{x}^T(t) \left(\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i + \mathbf{P} \mathbf{B}_i \mathbf{K}_j + \mathbf{K}_j^T \mathbf{B}_i^T \mathbf{P} + \mathbf{A}_p^T \mathbf{A}_p + (\mathbf{B}_p \mathbf{K}_j)^T (\mathbf{B}_p \mathbf{K}_j) + 2\mathbf{P} \mathbf{P} \right) \mathbf{x}(t) \right\} \\ &+ \left[\mathbf{x}^T(t) \mathbf{P} \mathbf{w}(t) + \mathbf{w}^T(t) \mathbf{P} \mathbf{x}(t) - \gamma^2 \mathbf{w}^T(t) \mathbf{w}(t) - \frac{1}{\gamma^2} \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) \right] \\ &+ \frac{1}{\gamma^2} \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) + \gamma^2 \mathbf{w}^T(t) \mathbf{w}(t) \end{aligned} \quad (5.42)$$

$$\begin{aligned} &= \sum_{i=1}^r \sum_{j=1}^r h_i(\mathbf{z}(t)) h_j(\mathbf{z}(t)) \\ &\left\{ \mathbf{x}^T(t) \left(\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i + \mathbf{P} \mathbf{B}_i \mathbf{K}_j + \mathbf{K}_j^T \mathbf{B}_i^T \mathbf{P} + \mathbf{A}_p^T \mathbf{A}_p + (\mathbf{B}_p \mathbf{K}_j)^T (\mathbf{B}_p \mathbf{K}_j) + 2\mathbf{P} \mathbf{P} \right) \mathbf{x}(t) \right\} \\ &+ \left(\frac{1}{\gamma} \mathbf{P} \mathbf{x}(t) - \gamma \mathbf{w}(t) \right)^T \left(\frac{1}{\gamma} \mathbf{P} \mathbf{x}(t) - \gamma \mathbf{w}(t) \right) \\ &+ \frac{1}{\gamma^2} \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) + \gamma^2 \mathbf{w}^T(t) \mathbf{w}(t) \\ &\leq \sum_{i=1}^r \sum_{j=1}^r h_i(\mathbf{z}(t)) h_j(\mathbf{z}(t)) \\ &\left\{ \mathbf{x}^T(t) \left(\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i + \mathbf{P} \mathbf{B}_i \mathbf{K}_j + \mathbf{K}_j^T \mathbf{B}_i^T \mathbf{P} + \mathbf{A}_p^T \mathbf{A}_p + (\mathbf{B}_p \mathbf{K}_j)^T (\mathbf{B}_p \mathbf{K}_j) + 2\mathbf{P} \mathbf{P} \right) \mathbf{x}(t) \right\} \\ &+ \frac{1}{\gamma^2} \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) + \gamma^2 \mathbf{w}^T(t) \mathbf{w}(t) \end{aligned} \quad (5.43)$$

from Equ. [5.41], we get,

$$\begin{aligned} &\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i + \mathbf{P} \mathbf{B}_i \mathbf{K}_j + \mathbf{K}_j^T \mathbf{B}_i^T \mathbf{P} + \mathbf{A}_p^T \mathbf{A}_p \\ &+ (\mathbf{B}_p \mathbf{K}_j)^T (\mathbf{B}_p \mathbf{K}_j) + 2\mathbf{P} \mathbf{P} < -\frac{1}{\gamma^2} \mathbf{P} \mathbf{P} - \mathbf{Q} \end{aligned} \quad (5.44)$$

from Equ. [5.43] and Equ. [5.44], we get,

$$\begin{aligned} \dot{V}(t) &< \sum_{i=1}^r \sum_{j=1}^r h_i(\mathbf{z}(t)) h_j(\mathbf{z}(t)) \mathbf{x}^T(t) \left[-\mathbf{Q} - \frac{1}{\gamma^2} \mathbf{P} \mathbf{P} \right] \mathbf{x}(t) \\ &+ \frac{1}{\gamma^2} \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) + \gamma^2 \mathbf{w}^T(t) \mathbf{w}(t) \end{aligned} \quad (5.45)$$

from the properties of $h_i(\mathbf{z}(t))$, the above inequality can imply the following inequality:

$$\begin{aligned}\dot{V}(t) &< -\mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) - \frac{1}{\gamma^2} \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) + \frac{1}{\gamma^2} \mathbf{x}^T(t) \mathbf{P} \mathbf{P} \mathbf{x}(t) + \gamma^2 \mathbf{w}^T(t) \mathbf{w}(t) \\ &= -\mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) + \gamma^2 \mathbf{w}^T(t) \mathbf{w}(t)\end{aligned}\quad (5.46)$$

Since $\|\mathbf{w}(t)\| \leq w_{bd}$, we get

$$\begin{aligned}\dot{V}(t) &< -\mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) + \gamma^2 w_{bd}^2 \\ &\leq -c_1 \mathbf{x}^T(t) \mathbf{x}(t) + \gamma^2 w_{bd}^2\end{aligned}\quad (5.47)$$

where $c_1 = \lambda_{\min}(\mathbf{Q})$ (the minimal eigenvalue of $\mathbf{Q}$), whenever $\|\mathbf{x}(t)\| > \gamma w_{bd} / (\sqrt{c_1})$, $\dot{V}(t) < 0$.

According to a standard LYAPUNOV extension, this demonstrates that the trajectories of the closed-loop system Equ. [5.24] are bounded.

Integrating Equ. [5.46] from $t = 0$ to $t = t_f$ yields,

$$\begin{aligned}V(t_f) - V(0) \\ < -\int_0^{t_f} \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) dt + \gamma^2 \int_0^{t_f} \mathbf{w}^T(t) \mathbf{w}(t) dt\end{aligned}\quad (5.48)$$

from Equ. [5.37], we get,

$$\int_0^{t_f} \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) dt < \mathbf{x}^T(0) \mathbf{P} \mathbf{x}(0) + \gamma^2 \int_0^{t_f} \mathbf{w}^T(t) \mathbf{w}(t) dt \quad (5.49)$$

this is Equ. [5.36] and the H^∞ control performance is achieved with a prescribed γ^2 .

In the case of $\mathbf{w}(t)$, if the fuzzy controller Equ. [5.24] is employed in the closed-loop nonlinear system Equ. [5.25] and there exists a positive-definite matrix $\mathbf{P} = \mathbf{P}^T > 0$ such that the matrix inequalities given in Equ. [5.41] are valid, this results in quadratically stable closed-loop system of Equ. [5.24].

5.2.7 H^∞ LMI Formulation

In general, it is not so obvious to find an analytical common solution in such away $\mathbf{P} = \mathbf{P}^T > 0$ for Equ. [5.41], especially in terms of solution of the Riccati equations. In this

sense, the synthesis problem is reformulated into the Linear Matrix Inequality problem (LMI). Hence, the matrix inequalities in Equ. [5.41] are transformed to the equivalent LMI's by introducing new variables $\mathbf{W}$ and $\mathbf{Y}_j$ in such a way $\mathbf{W} = \mathbf{P}^{-1}$ and $\mathbf{Y}_j = \mathbf{K}_j \mathbf{W}$, is equivalent to the following matrix inequalities:

$$\begin{aligned} & \mathbf{W} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{W} + \mathbf{B}_i \mathbf{Y}_j + \mathbf{Y}_j^T \mathbf{B}_i^T + \mathbf{W} \mathbf{A}_p^T \mathbf{A}_p \mathbf{W} \\ & + (\mathbf{B}_p \mathbf{Y}_j)^T (\mathbf{B}_p \mathbf{Y}_j) + \left(2 + \frac{1}{\gamma^2}\right) \mathbf{I} + \mathbf{W} \mathbf{Q} \mathbf{W} < 0 \end{aligned} \quad (5.50)$$

Which can be finally rewritten in a general form as,

$$\begin{bmatrix} \mathbf{W} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{W} + \mathbf{B}_i \mathbf{Y}_j + \mathbf{Y}_j^T \mathbf{B}_i^T + \left(2 + \frac{1}{\gamma^2}\right) \mathbf{I} & (\mathbf{B}_p \mathbf{Y}_j)^T & \mathbf{W} \\ \mathbf{B}_p \mathbf{Y}_j & -\mathbf{I} & 0 \\ \mathbf{W} & 0 & -\{\mathbf{A}_p^T \mathbf{A}_p + \mathbf{Q}\}^{-1} \end{bmatrix} < 0 \quad (5.51)$$

for $i, j = 1, 2, \dots, r$

If the LMI's in Equ. [5.51] have a positive-definite solution for $\mathbf{W}$, then the closed-loop system is stable and the H^∞ control performance in Equ. [5.36] is guaranteed for a prescribed γ .

The H^∞ optimization design for fuzzy control system of Equ. [5.2] is formulated as the following constrained optimization problem:

$$\text{Minimize } \gamma^2$$

$$\text{Subject to } \mathbf{W} = \mathbf{W}^T > 0$$

and

$$\begin{bmatrix} \mathbf{W} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{W} + \mathbf{B}_i \mathbf{Y}_j + \mathbf{Y}_j^T \mathbf{B}_i^T + \left(2 + \frac{1}{\gamma^2}\right) \mathbf{I} & (\mathbf{B}_p \mathbf{Y}_j)^T & \mathbf{W} \\ \mathbf{B}_p \mathbf{Y}_j & -\mathbf{I} & 0 \\ \mathbf{W} & 0 & -\{\mathbf{A}_p^T \mathbf{A}_p + \mathbf{Q}\}^{-1} \end{bmatrix} < 0 \quad (5.52)$$

where solutions for $\mathbf{W}$ and $\mathbf{Y}_j$ can be computed numerically and efficiently by convex optimization algorithm, already introduced in chapter two.

In reference to the LMI Toolbox of MATLAB [32] and by expressing Equ. [5.52] in to the standard form of Equ. [2.69] in chapter two, H^∞ controller $\mathbf{K}$ is synthesized.

Having derived the complete LMI formulation for the design of H^∞ fuzzy control of a nonlinear system in terms of its T-S models, the next section shows how can such LMI formulation can be used to synthesis a typical controller.

5.3 Case Studies on H^∞ Robust Control Synthesis

The last section has discussed the potential of designing a H^∞ robust regulator controller for a fuzzy system via a LMI formulation (Equ. [5.52]). This section will rather focus on some simulation design for both the nonlinear mass-spring-damper (SISO) system and the nonlinear antenna system (MIMO) presented in Appendices [A-1] and [A-2].

[A] Nonlinear Mass-Spring-Damper System

With reference to the nonlinear mass-spring-damper system presented in Appendix [A-1], this system has been used to design a robust H^∞ fuzzy controller as will be explained below.

Through the T-S fuzzy modeling procedure of the mass-spring-damper system in section 4.10, it was found that the system was sub-modeled as three linearized models over different operating regions. In this respect, the three sub-models are,

Sub-model 1:

$$\mathbf{A}_1 = \begin{bmatrix} -0.318 & -0.064 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 0.0123 \\ 0 \end{bmatrix}, \quad \mathbf{C}_1 = [1 \quad 0]$$

Sub-model 2:

$$\mathbf{A}_2 = \begin{bmatrix} -0.318 & 0.100 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0.100 \\ 0 \end{bmatrix}, \quad \mathbf{C}_2 = [1 \quad 0]$$

Sub-model 3:

$$\mathbf{A}_3 = \begin{bmatrix} -0.020 & 0.200 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B}_3 = \begin{bmatrix} 0.030 \\ 0 \end{bmatrix}, \quad \mathbf{C}_3 = [1 \quad 0]$$

The T-S fuzzy model exactly represents the nonlinear system, where such fuzzy models have common C and D matrices.

Open Loop Frequency Response

Once the linearized state-space sub-models are obtained, it is necessary to have a look at the frequency response before the design of the controller is discussed. In this respect, Figure [5.1] shows the open loop frequency response of a fuzzy sub-model 1, where it shows that the system does exhibit a fair degree of stability with short enough settling time. At high frequency, the system has a slow decrease in magnitude from 0.1 up to 1 rad/sec. In terms of the largest singular value of the system, Figure [5.2] shows the same result using singular value decomposition, where such largest singular value will be used to design the H^∞ controller.

Disturbance Rejection

As was mentioned already, we would like to design a robust H^∞ controller in such a way to have a fair disturbance rejection over a quite wide range of frequencies. In this respect, the H^∞ controller has been synthesized with the aid of LMI toolbox for each identified sub-model.

By representing each sub-model by the interconnection matrix $P(s)$ given in Appendix [A-1], the mass-spring fuzzy model interconnection matrix is given by,

$$P(s) = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}$$

where,

$$A = \begin{bmatrix} 1.2 & 2.3 \\ 1 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1.1 \\ 0 \end{bmatrix},$$

$$C_1 = [1 \ 0], \quad C_2 = [1 \ 0],$$

$$D_{11} = 0, \quad D_{12} = 0, \quad D_{21} = 0, \quad D_{22} = 0,$$

To synthesis the H^∞ fuzzy controller it is essential to determine three feedback gains in its consequent parts. After solving the LMIs formulated problem presented by Equ. [5.52], the feedback gains are obtained for each fuzzy sub-model, a typical fuzzy control law will be,

$$\text{IF } u(k) \text{ is } F_1^1 \text{ AND } y(k-1) \text{ is } F_2^1 \quad \text{THEN } u_1(t) = K_1 x(t)$$

In this respect, Figure [5.3] shows the impulse responses of the mass-spring system with the associated controller. It shows how the synthesized controller is able to regulate the system output in a robust way. This is done by bringing back the system output (mass position) to the zero state after the system has been subjected to a disturbance at its input.

Also Figure [5.3] shows the response of the system for different levels of disturbances, however, the controller was able to bring back the system output to the zero state within 0.01 seconds.

Closed Loop Frequency Response

Feedback gains have been used to form the closed loop system, hence for further examination, the closed loop frequency responses over a quite wide range of frequencies is shown in Figure [5.6]. For instance, $(T_o:Y_1)$ represents the relation between the disturbance w input and the z output signal. Since the mass-spring system has only one output, hence one Bode is shown i.e. $(T_o:Y_1)$. In the same graph, $(T_o:Y_2)$ describes the relation between the w disturbance input signal and the system output y .

Both $(T_o:Y_1)$ and $(T_o:Y_2)$ show a wide range of frequencies over which the two outputs have very low gain and constant behavior (up to 10^5 rad/sec). Hence emphasizing the low gain to eliminate the effect of disturbance.

For the purpose of investigating the closed loop frequency response of the designed controller (i.e. between y and w). Figure [5.7] shows the associated closed loop singular value of the system, which has been very well minimized to its lowest value over a wide range of frequencies i.e. from 0.1 rad/sec up to 10^4 rad/sec. From the figure as well, it is shown how the frequency response has got a nearly flat response over a wide range of frequency. The maximum magnitude of H^∞ norm is close to 0.000015 dB. In this respect the H^∞ norm has been achieved.

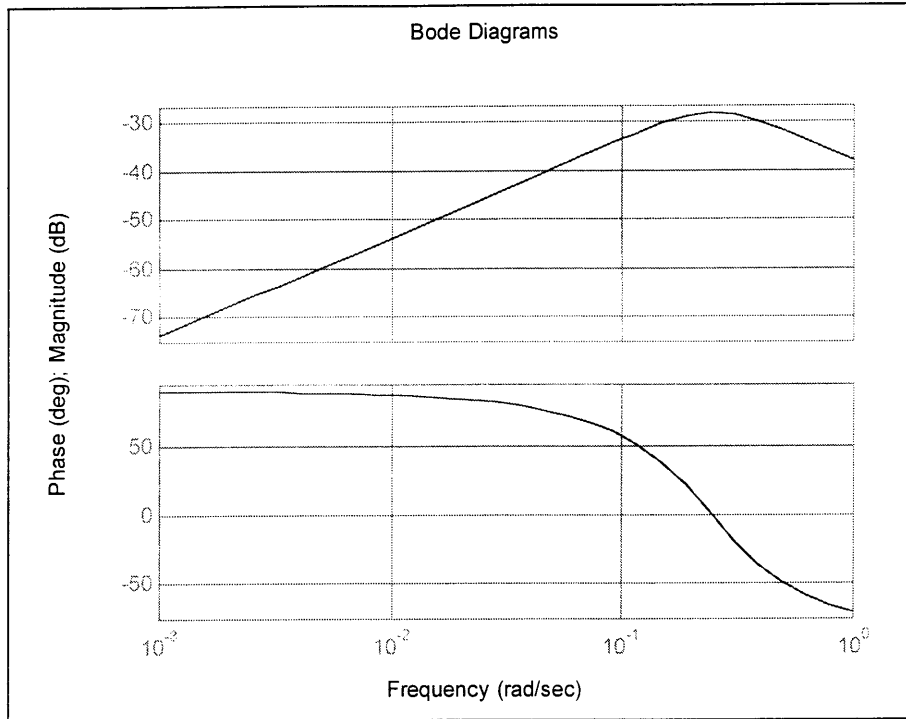


Figure [5.1] Bode plot of the open loop fuzzy sub-model 1 of the mass-spring system.

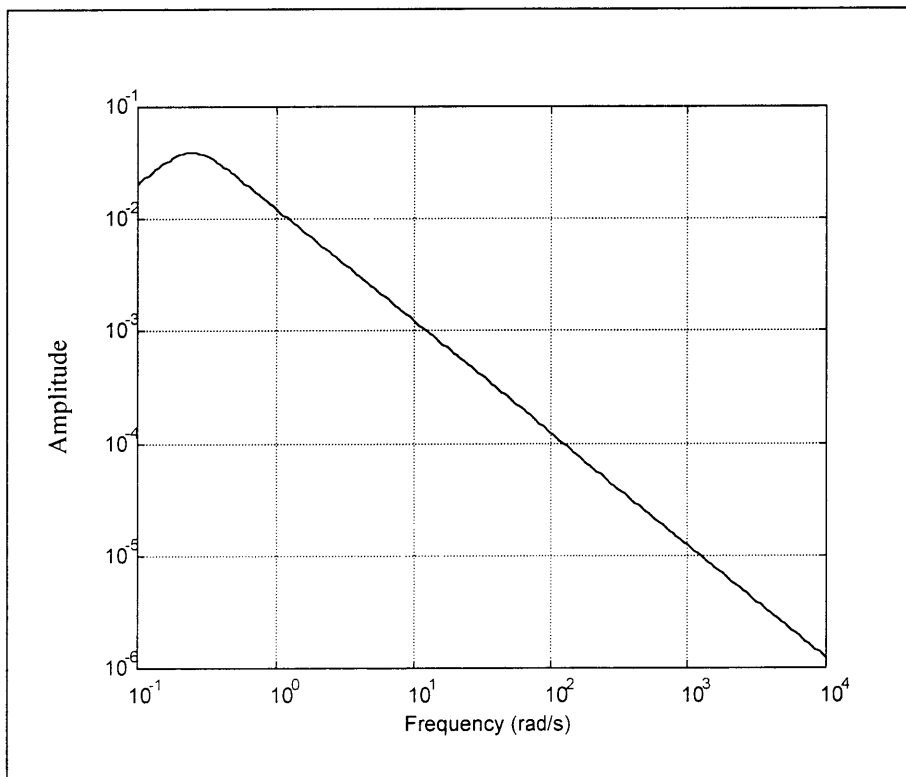


Figure [5.2] Maximum singular value of the open loop sub-model 1 of the mass-spring system.

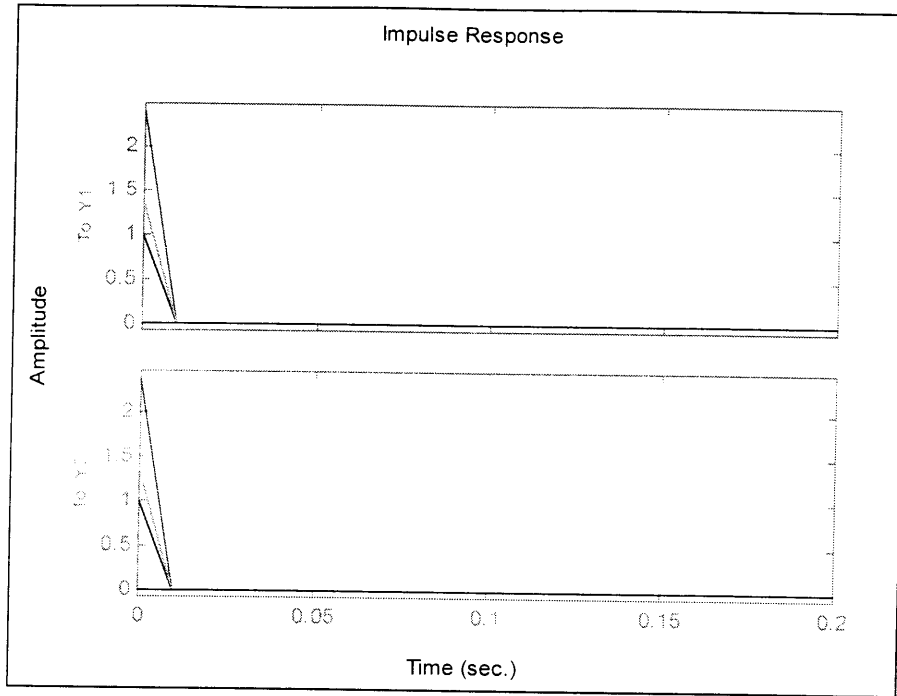


Figure [5.3] Impulse response of fuzzy model of the mass-spring system due to different disturbance levels where To:Y1 is the response between w and z, To:Y2 is between w and y.

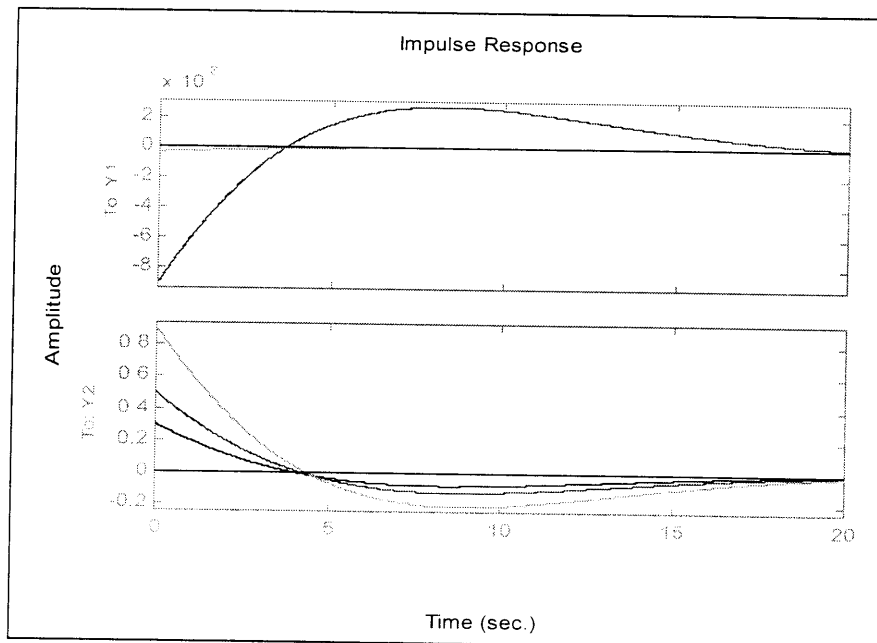


Figure [5.4] Impulse response of fuzzy sub-models of the mass-spring system due to different disturbance levels when $D_{12}=1$, where To:Y1 is the response between w and z, To:Y2 is between w and y.

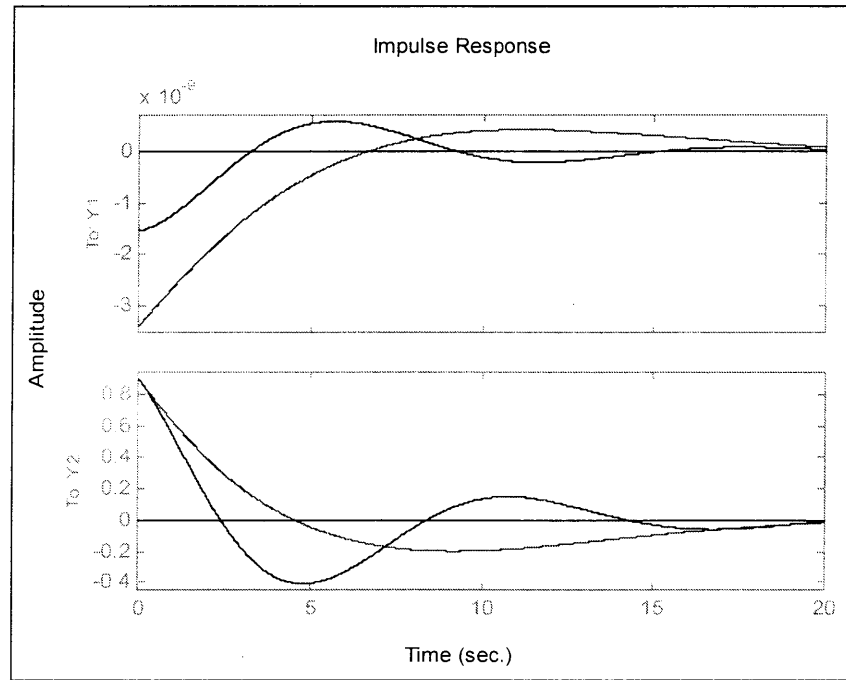


Figure [5.5] Impulse responses of fuzzy sub-model 1 & 2 of the mass-spring System when $D_{12}=1$, where To:Y1 is the response between w and z , To:Y2 is between w and y .

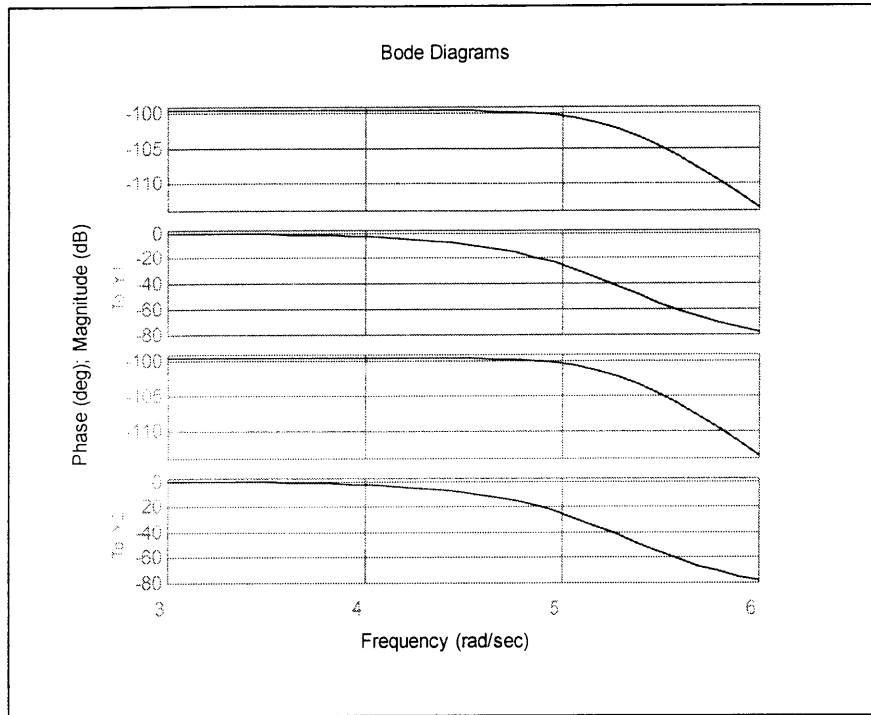


Figure [5.6] Frequency Response of the closed loop fuzzy sub-model 1 of the mass-spring system. (To:Y1 is the frequency response between z and w , To:Y2 is the frequency response between y and w).

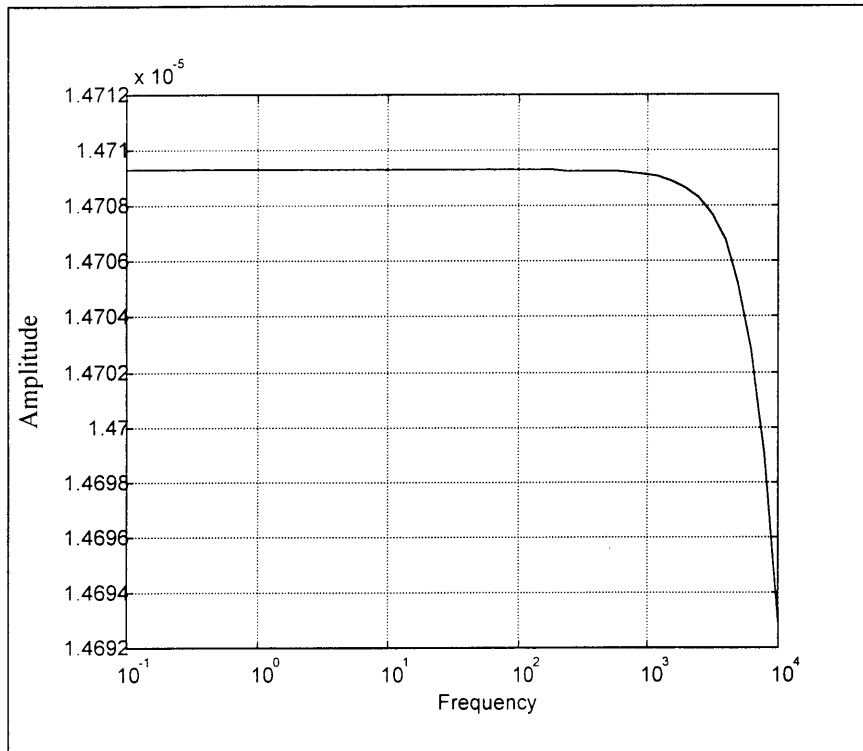


Figure [5.7] Maximum Singular value of the closed loop fuzzy sub-model 1 of the mass-spring system.

[B] The Nonlinear Antenna System

Through the T-S fuzzy modeling procedure of the antenna system already discussed in section 4.10, it was shown that the nonlinear antenna system was sub-modeled as three linearized models over the operating regions. In this respect, the three sub-models state-space matrices have been derived as,

$$\mathbf{A}_1 = \begin{bmatrix} 0.09510 & -0.4265 & 0.0300 & -0.2998 & 0.0116 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0.0348 & -0.0273 & 0.8022 & -0.3315 & 0 & 0.0013 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 0.0128 & 0 \\ 0 & 0 \\ 0 & 0.0240 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0.9354 & -0.4667 & 0.0525 & -0.0475 & 0.0119 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0.0373 & -0.0384 & 0.0844 & -0.3715 & 0 & -0.0002 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} 0.0128 & 0 \\ 0 & 0 \\ 0 & 0.0137 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 0.9020 & -0.4099 & 0.0142 & -0.0157 & 0.0108 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0.1294 & -0.1098 & 0.7887 & -0.2987 & 0 & 0.0002 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_3 = \begin{bmatrix} 0.0118 & 0 \\ 0 & 0 \\ 0 & 0.0028 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{D}_1 = \mathbf{D}_2 = \mathbf{D}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where the $\mathbf{C}$ and $\mathbf{D}$ matrix are common for all of the three fuzzy sub-models, and the $\mathbf{D}$ matrix is equal to zero.

Open Loop Frequency Response

Figure [5.8] displays the open loop frequency response of the antenna first fuzzy sub-model where it shows clearly that the modeled system is fairly stable for both outputs. In terms of

the singular values of the antenna fuzzy model, Figure [5.9] shows the frequency response of the open loop antenna system using singular value decomposition. The system two singular values are nearly constant over a wide frequency range (i.e. from 0.0001 rad/sec up to 1 rad/sec), then it starts to decrease with a nearly constant slope at higher frequencies.

Disturbance Rejection

The interconnection matrix $\mathbf{P}(s)$ of the antenna fuzzy model is given by,

$$\mathbf{P}(s) = \begin{bmatrix} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix}$$

where,

$$\mathbf{A}_1 = \begin{bmatrix} 0.09510 & -0.4265 & 0.0300 & -0.2998 & 0.0116 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0.0348 & -0.0273 & 0.8022 & -0.3315 & 0 & 0.0013 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0.0128 & 0 \\ 0 & 0 \\ 0 & 0.0240 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{C}_1 = [0 \ 0 \ 1 \ 0 \ 0 \ 0], \quad \mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{D}_{11} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{D}_{12} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{D}_{21} = 0, \quad \mathbf{D}_{22} = [0 \ 0]$$

The robust H^∞ fuzzy controller is to determine three feedback gains. After solving for the H^∞ gains using the LMIs presented in the last section, the feedback gains are obtained for each sub-model.

A typical fuzzy control law will be,

$$\begin{aligned} \text{IF } T_\varphi(k-1) \text{ is } F_1^1 \text{ AND } T_\varphi(k-2) \text{ is } F_2^1 \text{ AND } T_\psi(k-1) \text{ is } F_3^1 \text{ AND } \varphi(k-1) \text{ is } F_4^1 \\ \text{AND } \varphi(k-2) \text{ is } F_5^1 \text{ AND } \psi(k-1) \text{ is } F_6^1 \text{ AND } \psi(k-2) \text{ is } F_7^1 \\ \text{THEN } \mathbf{u}_1(t) = \mathbf{K}_1 \mathbf{x}(t) \end{aligned}$$

For evaluating the closed loop fuzzy control behavior, Figure [5.10] shows how the synthesized controller is able to regulate the system output in a robust way, by bringing back the system output to the zero state after the system has been subjected to a disturbance at its input. The three responses are for the antenna three outputs. The response is settled down in 15 seconds, which is a fast settling time according to the antenna system. In Figure [5.10], the first output $T_0:Y_1$ is the $\mathbf{z}$ output, whereas the second $T_0:Y_2$ and the third $T_0:y_3$ outputs represent the *azimuth* and *elevation* angles controlled outputs in the antenna system.

Closed Loop Frequency Response

The feedback gains are used to form the closed loop system, hence for further examination, the closed loop frequency responses over a quit wide range of frequencies is shown in Figure [5.11]. For instance, $T_0:Y_1$ represents the relation between the disturbance input $\mathbf{w}$ and the output $\mathbf{z}$, whereas, in the same graph, $T_0:Y_2$ and $T_0:Y_3$ describe the relation between the disturbance $\mathbf{w}$ and the *azimuth* angle and the *elevation* angle respectively. The frequency responses indicate how the system gains have been well reduced by the H^∞ synthesis, thus emphasizing the H^∞ behavior.

For the purpose of investigating the closed loop singular values of the designed controller, Figure [5.12] shows the associated closed loop singular value of the system. It is evident from this figure, how the frequency response has low gain over a wide range of frequency.

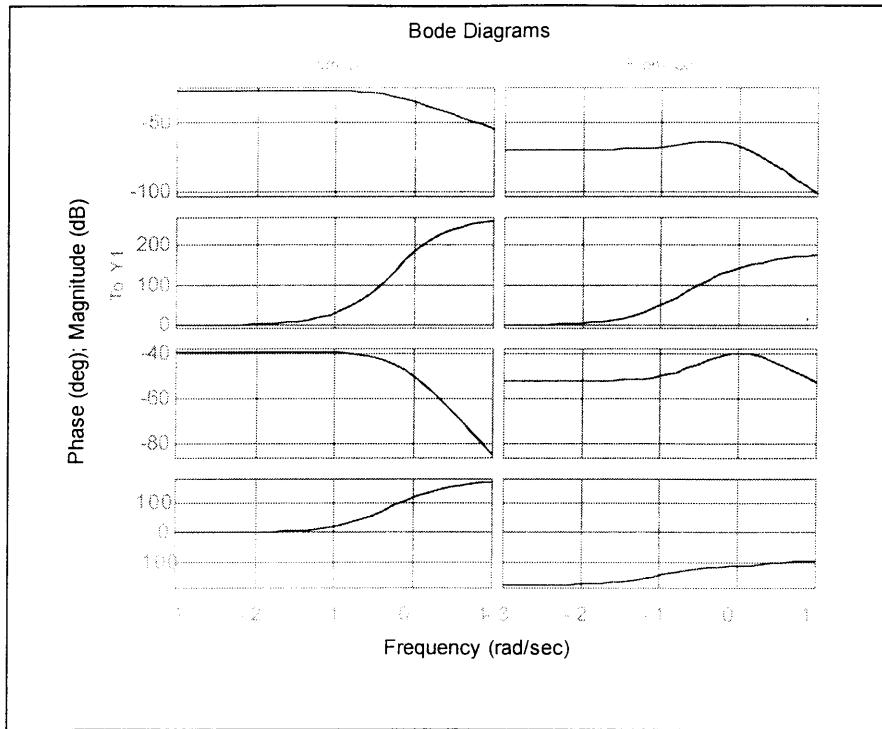


Figure [5.8] Frequency response of the open loop antenna fuzzy sub-model 1 for both inputs.

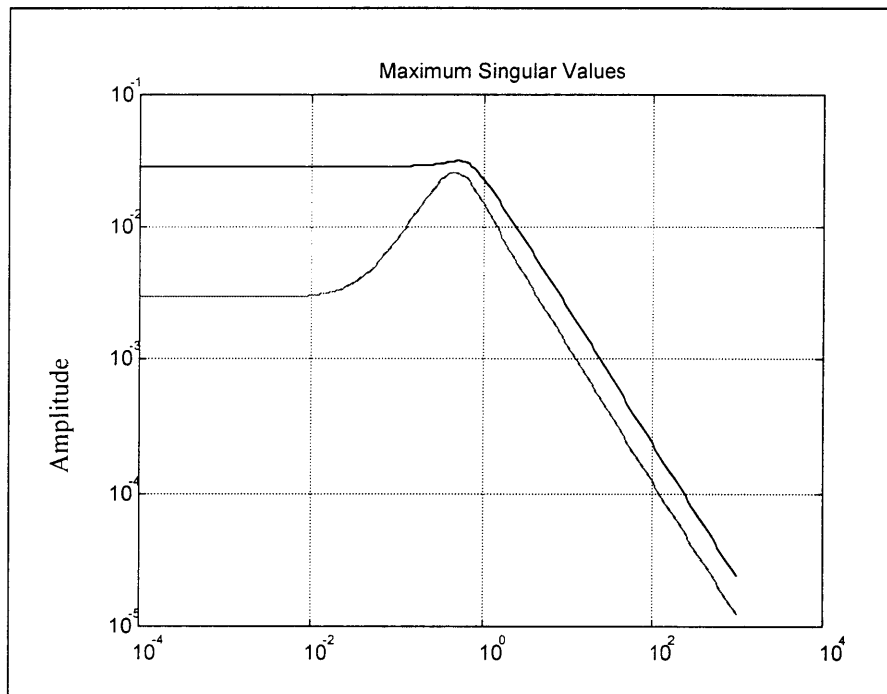


Figure [5.9] Maximum singular values of the open loop antenna fuzzy sub-model 1.

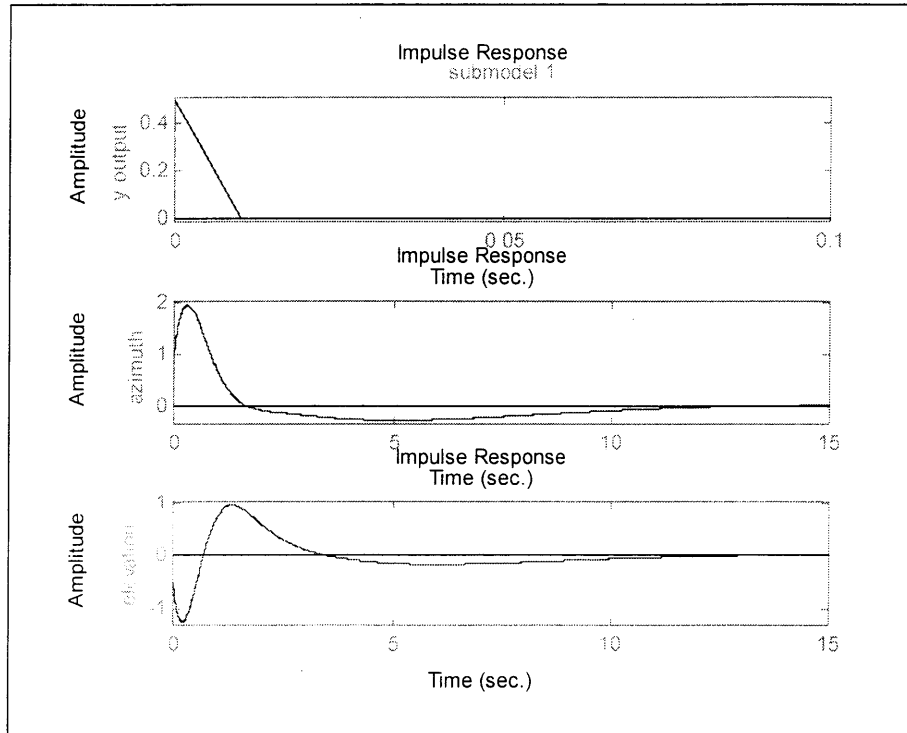


Figure [5.10] Impulse response of fuzzy model closed loop antenna system.

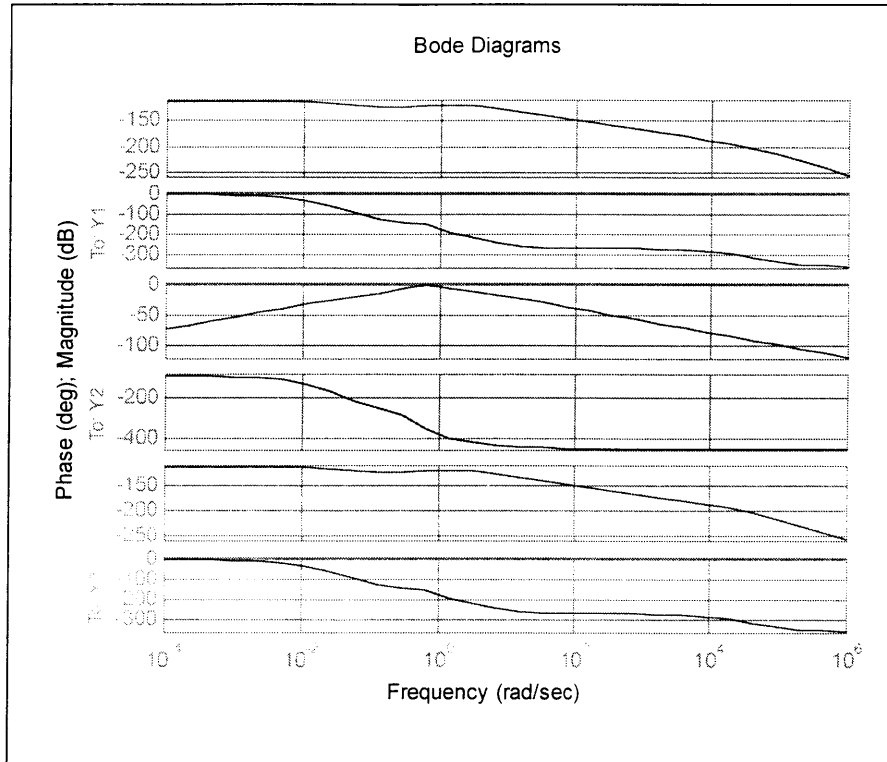


Figure [5.11] Frequency response of the controlled fuzzy sub-model 1 of the antenna system. (To:Y1 is the frequency response between z and w , To:Y2 is the frequency response between y and w).

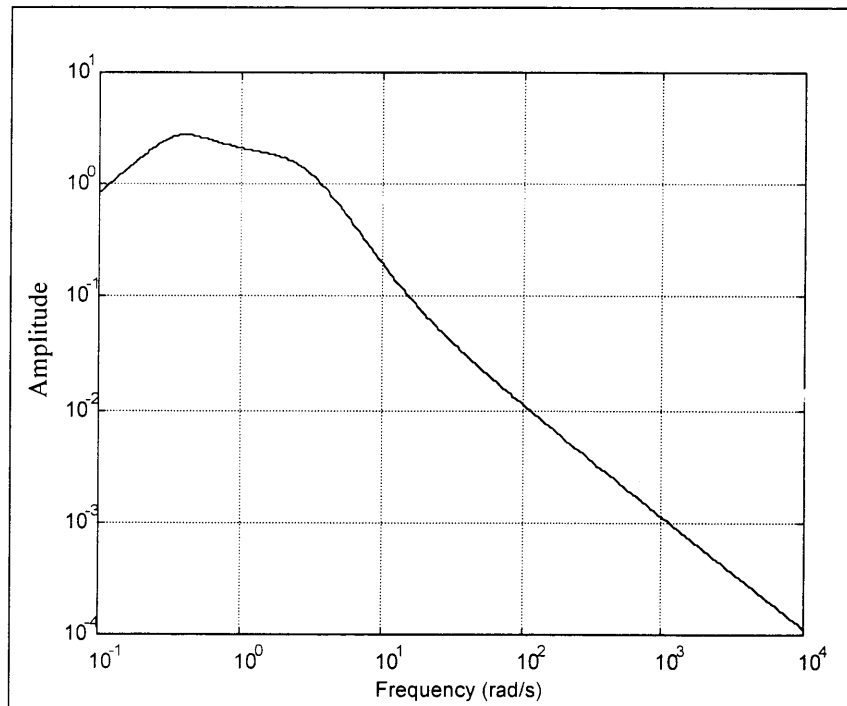


Figure [5.12] Maximum singular values of the controlled sub-model 1 of the antenna system.

5.4 Conclusions

In this chapter the bases for the robust fuzzy H^∞ controller design of a nonlinear system have been formulated and simulated for both a SISO and MIMO systems. Both systems do exhibit a degree of nonlinearity in terms of their dynamics, however such nonlinearities have been dealt with via the actual linearization by employment of Takagi-Sugeno modeling technique.

In term of the design of a robust H^∞ controller, the bases of design was achieved via derivation of the closed loop fuzzy control system, through which Lyapunov function has been derived and reduced to an LMI formulation for each sub-model of the nonlinear systems.

In terms of the LMI formulation, first a suitable Lyapunov function was chosen, from which the nonlinear closed loop stability was investigated. That was simply reduced to each sub-model due to the existing of different linear models representing the global nonlinear model.

In particular, the mass-spring system has been sub-modeled into three T-S models, through which stabilization controllers have been designed. The mass-spring system with a H^∞ controller has been simulated for an input disturbances. Once the controller was in the feedback, the mass-spring system was able to return to the original state, hence rejecting the disturbance effect.

With regard to the antenna system, H^∞ robust control system has been also synthesized in such a way the T-S sub-models have been utilized to synthesis the local controllers. The antenna response have been investigated and verified for some input disturbance, where the controller was able to bring back the antenna states back to the original states.

Chapter Six:

Fuzzy-Based H^∞ Robust Gain-Scheduled Controller

6.1 Introduction

The design of a robust H^∞ gain-scheduling is of essential importance once the system under concern exhibits specific changes in parameters due to the nonlinearity. Gain-scheduling problem has been a subject of a great deal of research over recent years, both from theoretical and practical point of views. This renewed interest stems from the development of new techniques and softwares which allow for a more rigorous and systematic treatment of the gain-scheduling problem. Expressed in terms of Linear Matrix Inequalities, the gain-scheduling problem is readily and globally solved using currently designed fuzzy system in addition to the available efficient LMI softwares.

Fuzzy H^∞ gain-scheduling is a new paradigm in which the system parameters are *extracted* from the *fuzzy models*, hence are used to synthesis a suitable controller. Changes in parameters from one fuzzy sub-model to another can be considered as the parameter dependent system, hence gain-scheduling control is synthesized in this respect. Gain-scheduler is achieved by expressing the fuzzy model of the system as Linear Parameter Varying (LPV) system, where H^∞ control is synthesized based on selecting a suitable feedback controller.

To verify the proposed methodology, parametric trajectories (indicating changes in system parameters as if they are in real time) have been generated in such a way to give a nonlinear parametric variations. At the end fuzzy H^∞ gain-scheduling for both the nonlinear mass-spring-damper system and the nonlinear antenna system have been achieved to verify the proposed methodology.

This chapter has been divided into a number of sections. Section 6.2, represents in a very primitive way the Takagi-Sugeno fuzzy models of nonlinear systems since they represent the bases for fuzzy gain-scheduling. Section 6.3 introduces Linear Parameter Varying (LPV) systems, whereas section 6.4 presents the H^∞ control for LPV systems. Section 6.5 discusses the mathematical formulation of H^∞ via LMI for LPV systems and section 6.6 and 6.7 present the employed fuzzy scheduler and the design of H^∞ fuzzy gain-scheduling. Case studies of employing gain-scheduling controller is presented in section 6.8. Finally section 6.9 draws few conclusions about the introduced H^∞ fuzzy gain-scheduler.

6.2 Takagi-Sugeno Fuzzy Model

Fuzzy controller design procedure is based on the representation of a given nonlinear plant in terms of the T-S fuzzy model given by Equ. [6.1]. As already introduced in chapter four, the

antecedent of each rule R_i contains fuzzy linguistic descriptions M_{ij} of the *scheduling variables* $\delta_j(t)$ and the *consequent* part contains a local linear sub-model of the nonlinear system. For instance, a typical rule with the scheduling variables $\delta(t)$ will be,

$$\begin{aligned} \text{Rule } i: \text{ IF } \delta_1(t) \text{ is } M_{i1} \dots \text{ AND } \delta_j(t) \text{ is } M_{ij} \\ \text{ THEN } \dot{\mathbf{x}}_i(t) = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t) + \mathbf{w}(t) \\ \mathbf{y}_i(t) = \mathbf{C}_i \mathbf{x}(t) \end{aligned} \quad (6.1)$$

The entire fuzzy model of the plant in Equ. [6.1] is obtained by blending of the consequent fuzzy sub-models. For a given pair of vectors $\mathbf{x}(t)$ and $\mathbf{u}(t)$, the final output of the fuzzy system is hence inferred as a weighted sum of the contributing sub-models:

$$\dot{\mathbf{x}}(t) = \frac{\sum_{i=1}^r \mu_i(\delta(t)) \{ \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t) \}}{\sum_{i=1}^r \mu_i(\delta(t))} \quad (6.2)$$

$$\mathbf{y}(t) = \frac{\sum_{i=1}^r \mu_i(\delta(t)) \{ \mathbf{C}_i \mathbf{x}(t) + \mathbf{D}_i \mathbf{u}(t) \}}{\sum_{i=1}^r \mu_i(\delta(t))} \quad (6.3)$$

where $\mu_i(\delta(t))$ is the *degree of fulfillment* of rule i , introduced already in chapter five.

Equ. [6.2] and Equ. [6.3] can be written as;

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^r h_i(\delta(t)) \{ \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t) \} \quad (6.4)$$

$$\mathbf{y}(t) = \sum_{i=1}^r h_i(\delta(t)) \{ \mathbf{C}_i \mathbf{x}(t) + \mathbf{D}_i \mathbf{u}(t) \} \quad (6.5)$$

While considering how fuzzy sub-models parameters vary from one sub-model to another, Eqs. [6.4-6.5] show that specific parameters in the obtained state-space model undergo variation over a specific range as the operating condition of the system changes.

The T-S fuzzy model can also be regarded as a quasi-linear system, i.e., a linear system in both $\mathbf{x}(t)$ and $\mathbf{u}(t)$ whose matrices $\mathbf{A}(\cdot)$, $\mathbf{B}(\cdot)$, $\mathbf{C}(\cdot)$ and $\mathbf{D}(\cdot)$ are not constant, but varying as the operating condition of the system changes. This is given as:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\delta(t))\mathbf{x}(t) + \mathbf{B}(\delta(t))\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(\delta(t))\mathbf{x}(t) + \mathbf{D}(\delta(t))\mathbf{u}(t)\end{aligned}\quad (6.6)$$

From Eqs. [6.2-6.3] one can observe that for all possible values of $\delta(t)$, these are bounded within a *polytope* whose vertices are the matrices of the individual rules.

The parameter dependence is affine; that is, the fuzzy state model $\mathbf{A}(\delta(t))$, $\mathbf{B}(\delta(t))$, $\mathbf{C}(\delta(t))$, $\mathbf{D}(\delta(t))$ depends affinely on $\delta(t)$. This means the time-varying parameter $\delta(t)$ varies in a polytope Θ of vertices $\delta_1, \delta_2, \dots, \delta_L$. Here the time varying parameter $\delta(t)$ is defined in terms of,

$$\delta(t) \in \Theta = Co\{\delta_1, \delta_2, \dots, \delta_L\}$$

where L is number of vertices in a polytope.

In other words, the state-space of Equ. [6.6] are given in terms of a number of models depending on $\delta(t)$,

$$\begin{bmatrix} \mathbf{A}(\delta(t)) & \mathbf{B}(\delta(t)) \\ \mathbf{C}(\delta(t)) & \mathbf{D}(\delta(t)) \end{bmatrix} \in Co\left\{ \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{bmatrix} : i = 1, \dots, L \right\} \quad (6.7)$$

where

$$Co\{\mathbf{S}_i : i = 1, 2, \dots, L\} = \left\{ \sum_{i=1}^L \alpha_i(t) \mathbf{S}_i : \sum_{i=1}^L \alpha_i(t) = 1, \alpha_i(t) \geq 0 \right\}$$

$$\text{and } \mathbf{S}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{bmatrix}$$

The described design method is intended for the control of nonlinear systems. In this case, the scheduling variables are a function of the state, hence $\delta(t) = \delta(\mathbf{x}(t))$ and $\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}(\mathbf{x})\mathbf{u}$.

Having presented T-S fuzzy model as a linear system with parametric changes, the next section presents the theory behind Linear Parameter Varying systems.

6.3 Linear Parameter Varying Systems (LPV)

There are three important classes of linear systems which are related to our investigations. The Linear Time-Invariant (LTI) systems is the more familiar class in the control system literature and give rise to the most successful and self-contained theories. LTI systems are described as,

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}\tag{6.8}$$

Secondly, Linear Time-Varying (LTV) systems, the state-space description of LTV systems is completely defined by the functional time-dependence of the state-space data $\mathbf{A}(t)$, $\mathbf{B}(t)$, $\mathbf{C}(t)$ and $\mathbf{D}(t)$ whose associated differential equations are,

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}(t)\mathbf{x} + \mathbf{B}(t)\mathbf{u} \\ \mathbf{y} &= \mathbf{C}(t)\mathbf{x} + \mathbf{D}(t)\mathbf{u}\end{aligned}\tag{6.9}$$

Finally, the Linear Parameter-Varying (LPV) systems, where the state-space model entries $\mathbf{A}(\cdot)$, $\mathbf{B}(\cdot)$, $\mathbf{C}(\cdot)$ and $\mathbf{D}(\cdot)$ are explicit functions of a time-varying parameter $\delta(t)$. Hence, they have a differential representation of the form of Equ. [6.10],

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}(\delta(t))\mathbf{x} + \mathbf{B}(\delta(t))\mathbf{u} \\ \mathbf{y} &= \mathbf{C}(\delta(t))\mathbf{x} + \mathbf{D}(\delta(t))\mathbf{u}\end{aligned}\tag{6.10}$$

LPV systems are “on-line” systems as they are completely known once the trajectory $\delta = \delta(t)$ is pre-defined, when the system is operating and experiences a particular trajectory in its domain.

In other words, the study of LPV systems requires the anticipation of the future behavior of the system and thus has immediate advantages for real-world applications.

The operating domain Θ of an LPV system Equ. [6.10] is the parameter range of the system, that is the domain of the parameter space where it takes its trajectories $\delta(t)$. Hence, LPV

system is well-defined whenever its parameter-dependence and its operating domain are fixed. This can be expressed more formally by the set of state-space relations as,

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}(\delta(t))\mathbf{x} + \mathbf{B}(\delta(t))\mathbf{u} \\ \mathbf{y} &= \mathbf{C}(\delta(t))\mathbf{x} + \mathbf{D}(\delta(t))\mathbf{u} \\ \delta(t) &\in \Theta, \quad \forall t \geq 0.\end{aligned}\tag{6.11}$$

LPV systems, can be viewed as linear systems subject to uncertainties $\delta(t)$. As a consequence an adequate controller $\mathbf{K}(s)$ (see Figure [6.1]) must be robust in the face of parameter uncertainties.

In addition, they can be viewed as a family of linear systems issuing from the linearization of a non-linear plant or more simply as parameter-dependent systems where the parameter values are directly exploitable in the control structure. This leads to a scheduling problem in a general sense as shown in Figure [6.2].

Mixing the above two situations; one comes up to the more general case where the parameter vector δ can be partitioned as $\delta = (\delta_1; \delta_2)^T$ where δ_1 is known in real-time and δ_2 is an uncertain parameter as shown in Figure [6.3].

The notation of LPV systems provides a formalization of various important issues of the control design problem. One of the most significant potentiality of the LPV framework is the derivation of LPV or “self-scheduled” controllers. Such controllers have the same parameter-dependence as the system and can be described in state-space form as

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}_K(\delta(t))\mathbf{x} + \mathbf{B}_K(\delta(t))\mathbf{y} \\ \mathbf{u} &= \mathbf{C}_K(\delta(t))\mathbf{x} + \mathbf{D}_K(\delta(t))\mathbf{y}\end{aligned}\tag{6.12}$$

6.4 Robust H^∞ Control of Linear Parametric Varying (LPV) Systems

For H^∞ design of linear parametric varying systems, it is essential that the system model is presented as LPV in the form of Equ. [6.10].

We are interested in a class of LPV systems where,

- (i) The parameter-dependence is affine, that is, the state-space matrices $\mathbf{A}(\delta(t))$, $\mathbf{B}(\delta(t))$, $\mathbf{C}(\delta(t))$, $\mathbf{D}(\delta(t))$ depend affinely on $\delta(t)$.

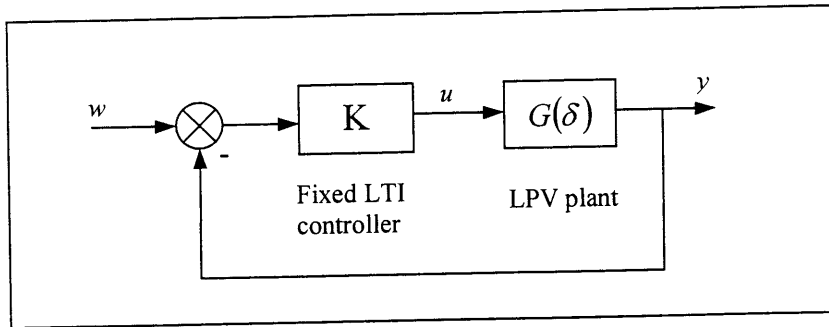


Figure [6.1] LTI Control of a LPV system.

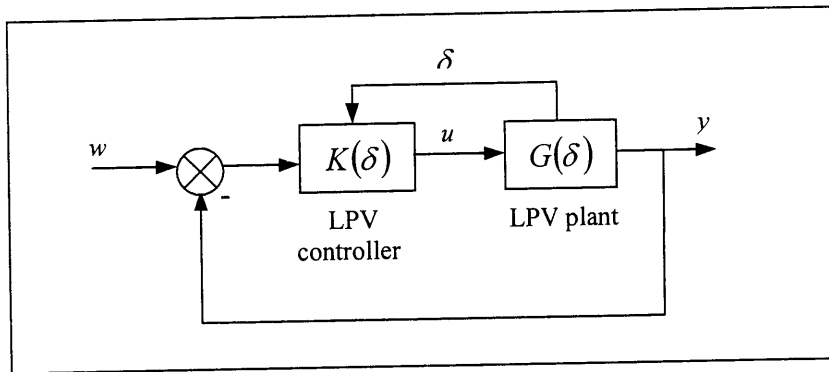


Figure [6.2] LPV Control of a LPV system.

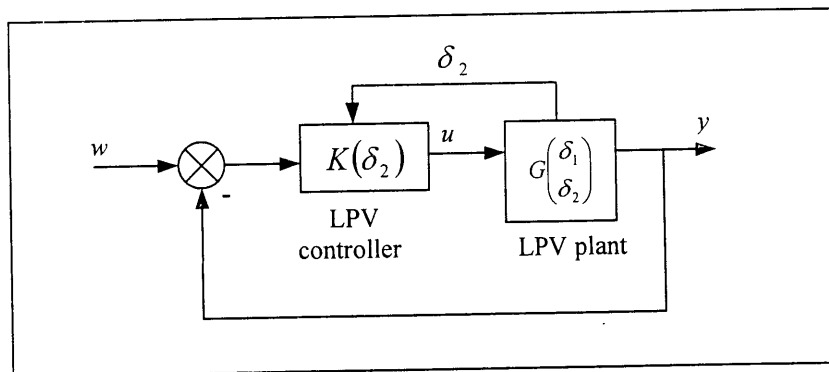


Figure [6.3] LPV Control of a mixed LPV system.

(ii) The time-varying parameter $\delta(t)$ varies in a polytope Θ of vertices $\delta_1, \delta_2, \dots, \delta_L$. That is,

$$\delta(t) \in \Theta := Co\{\delta_1, \delta_2, \dots, \delta_L\} \quad (6.13)$$

these vertices represent the extremal values of the parameters.

This description encompasses many practical situations. From this characterization, it is clear that the state-space matrices $\mathbf{A}(\delta(t))$, $\mathbf{B}(\delta(t))$, $\mathbf{C}(\delta(t))$, $\mathbf{D}(\delta(t))$ involved in a polytope of matrices whose vertices are the images of the vertices $\delta_1, \delta_2, \dots, \delta_L$.

In other words our fuzzy model is put as,

$$\begin{bmatrix} \mathbf{A}(\delta) & \mathbf{B}(\delta) \\ \mathbf{C}(\delta) & \mathbf{D}(\delta) \end{bmatrix} \in Co\left\{ \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{bmatrix} : i = 1, \dots, L \right\} \quad (6.14)$$

where

$$\begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\delta_i) & \mathbf{B}(\delta_i) \\ \mathbf{C}(\delta_i) & \mathbf{D}(\delta_i) \end{bmatrix} \quad (6.15)$$

Hence, the H^∞ control problem for LPV systems is similar to the customary H^∞ synthesis where the robust H^∞ norm bound is replaced by the more suitable notation of quadratic H^∞ performance and the plant under consideration is now LPV.

Hence, for LPV system defined by the state-space equation as,

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}(\delta(t))\mathbf{x} + \mathbf{B}(\delta(t))\mathbf{u} \\ \mathbf{y} &= \mathbf{C}(\delta(t))\mathbf{x} + \mathbf{D}(\delta(t))\mathbf{u} \end{aligned} \quad (6.16)$$

has quadratic H^∞ Performance γ if and only if there exists a Lyapunov function $V(\mathbf{x})$ such that $V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}$ with $\mathbf{P} > 0$ that establishes global stability and the L_2 gain of the input/output map is bounded by γ . That is,

$$\|\mathbf{y}\|_2 < \gamma \|\mathbf{u}\|_2 \quad (6.17)$$

along all possible parameter trajectories $\delta(t)$ in Θ .

LINEAR FRACTIONAL TRANSFORMATION FOR LPV SYSTEMS

An augmented LPV plant need to be considered [54], mapping exogenous inputs $\mathbf{w}$ and control inputs $\mathbf{u}$ to the controlled outputs $\mathbf{z}$ and measured outputs $\mathbf{y}$ as shown in Figure [6.4]. Such formulation is given by the standard Linear Fractional Transformation (LFT) already presented in chapter two (section 2.3) and given in terms of LPV as,

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}(\delta(t))\mathbf{x} + \mathbf{B}_1(\delta(t))\mathbf{w} + \mathbf{B}_2(\delta(t))\mathbf{u} \\ \mathbf{z} &= \mathbf{C}_1(\delta(t))\mathbf{x} + \mathbf{D}_{11}(\delta(t))\mathbf{w} + \mathbf{D}_{12}(\delta(t))\mathbf{u} \\ \mathbf{y} &= \mathbf{C}_2(\delta(t))\mathbf{x} + \mathbf{D}_{21}(\delta(t))\mathbf{w} + \mathbf{D}_{22}(\delta(t))\mathbf{u}\end{aligned}\quad (6.18)$$

$$\delta(t) \in \Theta := \text{Co}\{\delta_1, \delta_2, \dots, \delta_L\}, \quad \text{for } \forall t \geq 0$$

The plant is further assumed to be polytopic, i.e.

$$\begin{bmatrix} \mathbf{A}(\delta(t)) & \mathbf{B}_1(\delta(t)) & \mathbf{B}_2(\delta(t)) \\ \mathbf{C}_1(\delta(t)) & \mathbf{D}_{11}(\delta(t)) & \mathbf{D}_{12}(\delta(t)) \\ \mathbf{C}_2(\delta(t)) & \mathbf{D}_{21}(\delta(t)) & \mathbf{D}_{22}(\delta(t)) \end{bmatrix} \in \mathbf{P} := \text{Co}\left\{ \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_{1i} & \mathbf{B}_{2i} \\ \mathbf{C}_{1i} & \mathbf{D}_{11i} & \mathbf{D}_{12i} \\ \mathbf{C}_{2i} & \mathbf{D}_{21i} & \mathbf{D}_{22i} \end{bmatrix}, \quad i = 1, 2, \dots, L \right\} \quad (6.19)$$

in which $\mathbf{A}_i, \mathbf{B}_{1i}, \dots$ denote the values of $\mathbf{A}(\delta(t)), \mathbf{B}_1(\delta(t)), \dots$ at the vertices $\delta(t) = \delta_i$ of the parameter polytope. With reference to Equ. [6.19] the problem dimensions are thus given by

$$\mathbf{A}(\delta(t)) \in \mathbb{R}^{n \times n}, \quad \mathbf{D}_{11}(\delta(t)) \in \mathbb{R}^{p_1 \times m_1}, \quad \mathbf{D}_{22}(\delta(t)) \in \mathbb{R}^{p_2 \times m_1}$$

where the following assumptions on the plant are considered:

(i) $\mathbf{D}_{22}(\delta(t)) = 0$ or equivalently $\mathbf{D}_{22i} = 0$ for $i = 1, 2, \dots, L$.

(ii) $\mathbf{B}_2(\delta(t)), \mathbf{C}_2(\delta(t)), \mathbf{D}_{12}(\delta(t)), \mathbf{D}_{21}(\delta(t))$ are parameter-independent or equivalently,

$$\mathbf{B}_{2i} = \mathbf{B}_2, \quad \mathbf{C}_{2i} = \mathbf{C}_2, \quad \mathbf{D}_{12i} = \mathbf{D}_{12}, \quad \mathbf{D}_{21i} = \mathbf{D}_{21} \quad (i = 1, 2, \dots, L)$$

(iii) The pairs $(\mathbf{A}(\delta), \mathbf{B}_2)$ and $(\mathbf{A}(\delta), \mathbf{C}_2)$ are quadratically stabilizable and detectable respectively over Θ .

With such notations and assumptions, the H^∞ control problem for LPV systems can be stated as follows;

Synthesis a LPV robust controller of state-space form of,

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}_K(\delta(t))\mathbf{x} + \mathbf{B}_K(\delta(t))\mathbf{u} \\ \mathbf{y} &= \mathbf{C}_K(\delta(t))\mathbf{x} + \mathbf{D}_K(\delta(t))\mathbf{u}\end{aligned}\quad (6.20)$$

which guarantees Quadratic H^∞ Performance γ for the closed-loop system of Figure [6.4], in such away to ensure the following control objectives,

- The closed-loop system is quadratically stable over Θ ,
- The L_2 -induced of the operator mapping $\mathbf{w}$ into $\mathbf{z}$ bounded by γ for all possible trajectories $\delta(t)$ in Θ .

To find the controller of Equ. [6.20], the LMI technique has been used in this respect as will be presented in section 6.5.

6.5 LMI Formulation of LPV H^∞ Controller

Considering a continuous LPV polytopic system of Equ. [6.18] and working under the three assumptions in section 6.4, and by letting $\mathbf{N}_R$ and $\mathbf{N}_S$ denote bases of the null space of $(\mathbf{B}_2^T, \mathbf{D}_{12}^T)$ and $(\mathbf{C}_2, \mathbf{D}_{21})$, respectively, the H^∞ controller can be synthesized. There exists an LPV controller guaranteeing Quadratic H^∞ Performance γ along all parameter trajectories in the polytope Θ [32]. This is achieved if and only if there exist two symmetric matrices $(\mathbf{R}, \mathbf{S})$ in $\mathcal{R}^{n \times n}$ satisfying the system of $(2r+1)$ LMIs formulation of,

$$\begin{bmatrix} \mathbf{N}_R & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_i \mathbf{R} + \mathbf{R} \mathbf{A}_i^T & \mathbf{R} \mathbf{C}_{1i}^T & \mathbf{B}_{1i} \\ \mathbf{C}_{1i} \mathbf{R} & -\gamma \mathbf{I} & \mathbf{D}_{11i} \\ \mathbf{B}_{1i}^T & \mathbf{D}_{11i}^T & -\gamma \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{N}_R & 0 \\ 0 & \mathbf{I} \end{bmatrix} < 0 \quad i = 1, 2, \dots, L \quad (6.21)$$

$$\begin{bmatrix} \mathbf{N}_S & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_i \mathbf{S} + \mathbf{S} \mathbf{A}_i^T & \mathbf{S} \mathbf{C}_{1i}^T & \mathbf{B}_{1i} \\ \mathbf{C}_{1i} \mathbf{S} & -\gamma \mathbf{I} & \mathbf{D}_{11i} \\ \mathbf{B}_{1i}^T & \mathbf{D}_{11i}^T & -\gamma \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{N}_S & 0 \\ 0 & \mathbf{I} \end{bmatrix} < 0 \quad i = 1, 2, \dots, L \quad (6.22)$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{I} \\ \mathbf{I} & \mathbf{S} \end{bmatrix} \geq 0 \quad (6.23)$$

where r is the number of vertices.

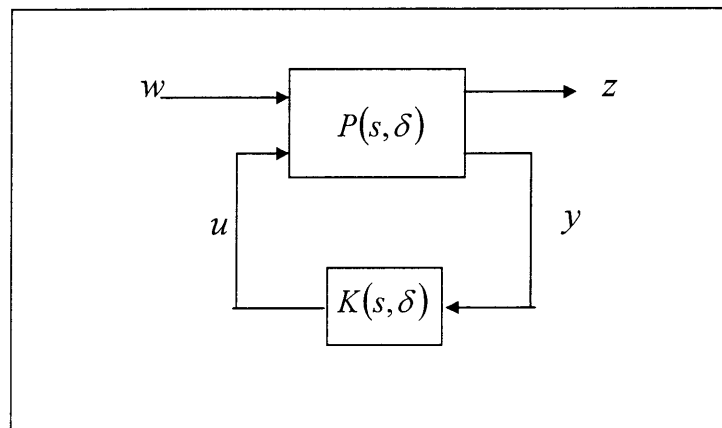


Figure [6.4] H^∞ Synthesis LFT structure for LPV systems.

Since these conditions are LMI's in the variables $\mathbf{R}$ and $\mathbf{S}$, they are convex and fall into the scope of efficient convex optimization techniques. For γ , $\mathbf{R}$ and $\mathbf{S}$ solutions to the LMI's Eqs. [6.21-6.23], there always exist LPV polytopic controllers solving the problem [55].

In turn, such controllers are described by a system of LMI's from which one can extract a particular solution by algebraic manipulations. More precisely, along some trajectory $\delta(t)$ in the polytope Θ , i.e.

$$\delta(t) = \sum_{i=1}^r \alpha_i(t) \delta_i \quad (6.24)$$

The state-space matrices $\mathbf{A}_K(\delta(t))$, $\mathbf{B}_K(\delta(t))$, $\mathbf{C}_K(\delta(t))$, $\mathbf{D}_K(\delta(t))$ of the LPV polytopic controllers are then read as,

$$\begin{bmatrix} \mathbf{A}_K(\delta(t)) & \mathbf{B}_K(\delta(t)) \\ \mathbf{C}_K(\delta(t)) & \mathbf{D}_K(\delta(t)) \end{bmatrix} = \sum_{i=1}^r \alpha_i(t) \begin{bmatrix} \mathbf{A}_{Ki} & \mathbf{B}_{Ki} \\ \mathbf{C}_{Ki} & \mathbf{D}_{Ki} \end{bmatrix} \quad (6.25)$$

where the α_i 's are computed according to the convex decomposition Equ. [6.24].

It is important to note that while the state-space data matrices $\mathbf{A}_{Ki}$, $\mathbf{B}_{Ki}$, $\mathbf{C}_{Ki}$, $\mathbf{D}_{Ki}$ are computed off-line, the LPV controller matrices $\mathbf{A}_K(\delta)$, $\mathbf{B}_K(\delta)$, $\mathbf{C}_K(\delta)$, $\mathbf{D}_K(\delta)$ must be updated in real time depending on the parameter measurement $\delta(t)$.

6.6 Fuzzy Controller

The control of nonlinear system by means of fuzzy gain-scheduler controller begins with the determination of the linear sub-models in the operating regions of interest of the system to be controlled. Optimization techniques are then used to design local controllers within a fuzzy gain-scheduling scheme with the desired overall behavior. In this way, a wide-range stabilization and control problems can be solved. For a reference input $\mathbf{r}$ and in the continuous-time case, the T-S fuzzy control rules have the form of,

$$\begin{aligned} \text{Rule } i: \quad & \text{IF } \delta_1(t) \text{ is } M_{i1} \dots \text{AND } \delta_j(t) \text{ is } M_{ij} \\ & \text{THEN } \mathbf{u}_i(t) = \mathbf{r}(t) - \mathbf{y}(t) - \mathbf{K}_i \mathbf{x}(t) \end{aligned} \quad (6.26)$$

where $\mathbf{K}$ is the controller gain and $\delta_1(t)$ is the scheduling variable. With $\mathbf{r}(t)$ as the reference, the controller's output can be written as,

$$\mathbf{u}(t) = \frac{\sum_{i=1}^r \mu_i(\delta(t)) \{\mathbf{r}(t) - \mathbf{y}(t) - \mathbf{K}_i \mathbf{x}(t)\}}{\sum_{i=1}^r \mu_i(\delta(t))} \quad (6.27)$$

and can be expressed as,

$$\begin{aligned} \mathbf{u}(t) &= \sum_{i=1}^r h_i(\delta(t)) \{\mathbf{r}(t) - \mathbf{y}(t) - \mathbf{K}_i \mathbf{x}(t)\} \\ &= \mathbf{r}(t) - [\mathbf{K}(\delta(t)) + \mathbf{C}(\delta(t))] \mathbf{x}(t) \end{aligned} \quad (6.28)$$

If the scheduling vector $\delta(t)$ is a function of the state $\mathbf{x}(t)$, then $\mathbf{u}(t)$ represents a nonlinear gain-scheduled control law. Hence, the goal of the controller design is to determine the constant matrix $\mathbf{K}_i$ such that the desired dynamic of the closed-loop system and the desired steady-state, input-output behavior are obtained. Designing the state-feedback gains $\mathbf{K}_i$ requires dealing with the system dynamics and hence ensuring stability. This problem is solved by means of LMI. For the T-S fuzzy controller in Equ. [6.26], a feedforward gains $\mathbf{V}_i$ can be also included in the design and can be given as,

$$\mathbf{V}_i = \left(\mathbf{C}_i (-\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_i)^{-1} \mathbf{B}_i \right)^{-1} \quad (6.29)$$

This ensures for each closed-loop sub-system a unity steady-state gain. In other words, if a perfect model is available, the T-S fuzzy controller Equ. [6.26] used for tracking yields $\mathbf{y}(t)$ approaches $\mathbf{r}(t)$. However, a reasonable requirement for the controller is rather to satisfy $\mathbf{x}(t)$ approaches 0 once t approaches ∞ . This implies the stabilization problem of the control system once $\mathbf{r}(t)=0$. Usually, this T-S fuzzy controller cannot be satisfactorily in tracking control problems where a given reference trajectory $\mathbf{r}(t) \neq 0$ is to be followed. The reason for this is the ever-present mismatch (in practice) between the fuzzy model and the real plant dynamic resulting in steady-state errors. A fuzzy scheduler which tackles this problem will be introduced later in section 6.7.

FUZZY CLOSED-LOOP DYNAMICS

The closed-loop system consisting of the fuzzy model and the fuzzy controller is obtained by substituting the fuzzy controller Equ. [6.28] into the state equation of the fuzzy model Equ. [6.4]. The closed-loop system is then given by,

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^r \sum_{j=1}^r h_i(\delta(t)) h_j(\delta(t)) \{(\mathbf{A}_i - \mathbf{B}_i(\mathbf{C}_i + \mathbf{K}_i))\mathbf{x}(t) + \mathbf{B}_i \mathbf{r}(t)\} \quad (6.30)$$

It is assumed throughout this section that; the weight of each rule in the fuzzy controller is equal to that of the corresponding rule in the fuzzy model. This assumption is easy to satisfy since all weighting factors of the controller can be simply taken over from the known fuzzy model. Then, Equ. [6.30] can be rewritten as,

$$\begin{aligned} \dot{\mathbf{x}}(t) = & \sum_{i=1}^r h_i(\delta(t)) h_i(\delta(t)) \mathbf{G}_{ii} \mathbf{x}(t) + 2 \sum_{i < j}^r h_i(\delta(t)) h_j(\delta(t)) \frac{\mathbf{G}_{ij} + \mathbf{G}_{ji}}{2} \mathbf{x}(t) \\ & + \sum_{i=1}^r \sum_{j=1}^r h_i(\delta(t)) h_j(\delta(t)) \mathbf{B}_i \mathbf{r}(t) \end{aligned} \quad (6.31)$$

where

$$\mathbf{G}_{ij} = (\mathbf{A}_i - \mathbf{B}_i \mathbf{K}_j) \quad (6.32)$$

For the particular case of common matrices $\mathbf{B}_i$, i.e., $\mathbf{B}_i = \mathbf{B}$ for all sub-models $i=1,2,\dots,r$ and for the shared rules, the following simplified description of the entire closed loop system can be derived as,

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^r h_i(\delta(t)) \{(\mathbf{A}_i - \mathbf{B}_i \mathbf{K}_i) \mathbf{x}(t) + \mathbf{B}_i \mathbf{r}(t)\} \quad (6.33)$$

Equ. [6.33] represents the terms for achieving closed loop stability for each fuzzy sub-model. They are responsible for the stability of the control system matrices $\mathbf{K}_j$ and calculated via LMIs such that a Quadratic LYAPUNOV function is found (as was presented in chapter five).

6.7 Fuzzy H^∞ Gain-Scheduler

With reference to Figure [6.5], the system receives input signal $\mathbf{u}(t)$, which has already computed based on the controller gain $\mathbf{K}$. Hence, it will be necessary to choose the appropriate value of $\mathbf{K}_i$ in accordance to the condition under which the system is operating.

In section 6.6 we introduced the H^∞ control under which the system to be controlled described as LPV system. Fuzzy gain-scheduling is introduced here in twofold,

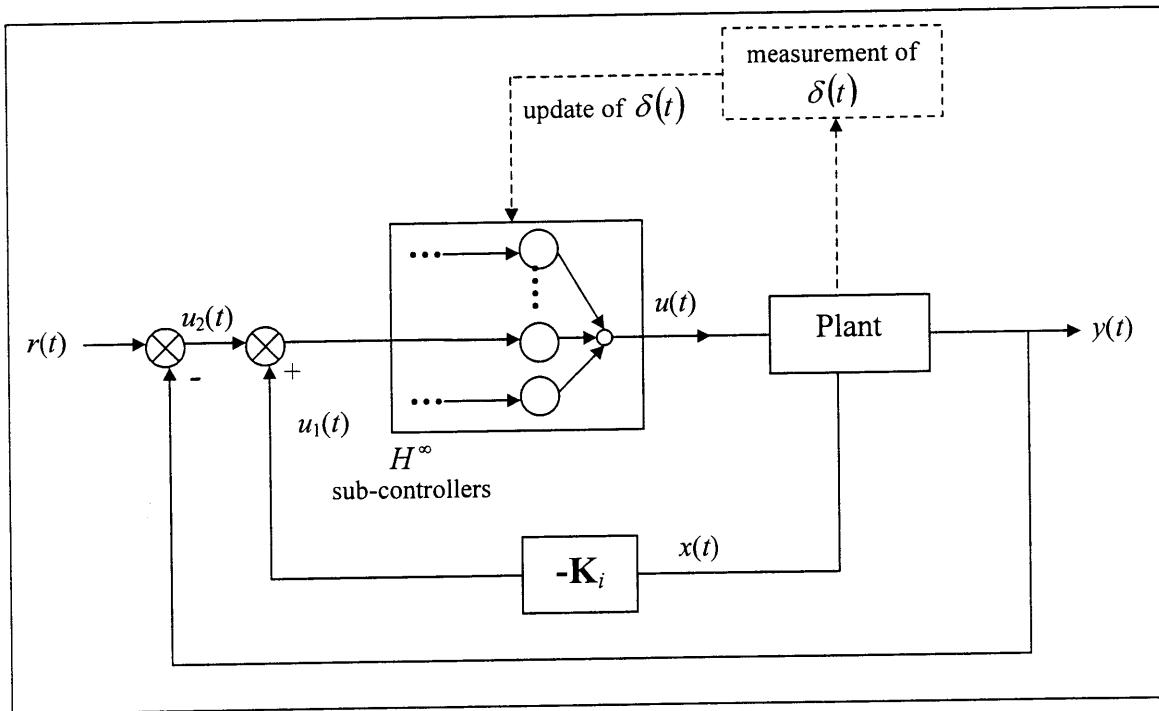


Figure [6.5] A dynamic Output Feedback H^∞ Fuzzy Controller.

The *first* is via the use of different fuzzy state space sub-models (i.e. T-S models) to designate the parameters that are changing. This will formulate the bases of constructing a suitable LPV system based on fuzzy models for the entire operating region. H^∞ controller is then synthesized in according to LPV systems.

The *second* of is to employ the LPV already obtained from fuzzy modeling in designing H^∞ controller $\mathbf{K}(s)$, hence, adapting the controller gain $\mathbf{K}$ using fuzzy IF-THEN rules. In this way, the system is getting much transparent and knowledgeable in terms of its operating condition.

The entire fuzzy system can then be described as,

$$\dot{\mathbf{x}}(t) = \mathbf{A}_w(\cdot)\mathbf{x}(t) + \mathbf{B}_w(\cdot)\mathbf{r}(t) \quad (6.34)$$

$$\mathbf{y}_w(t) = \mathbf{C}_w(\cdot)\mathbf{x}(t) \quad (6.35)$$

Based on the LPV theory already introduced in section 6.3, the fuzzy state-space matrices are given by,

$$\mathbf{A}_w(\cdot) = \sum_{i=1}^r \sum_{j=1}^r h_i(\delta(t))h_j(\delta(t))[\mathbf{A}_i - \mathbf{B}_i(\mathbf{C}_i + \mathbf{K}_j)] \quad (6.36)$$

$$\mathbf{B}_w(\cdot) = \sum_{i=1}^r h_i(\delta(t))\mathbf{B}_i \quad (6.37)$$

$$\mathbf{C}_w(\cdot) = \sum_{i=1}^r h_i(\delta(t))\mathbf{C}_i \quad (6.38)$$

These equations are expressed as,

$$\mathbf{A}_w(\cdot) = \sum_{i=1}^r h_i(\delta(t))\mathbf{A}_w^i \quad (6.39)$$

$$\mathbf{B}_w(\cdot) = \sum_{i=1}^r h_i(\delta(t))\mathbf{B}_w^i \quad (6.40)$$

$$\mathbf{C}_w(\cdot) = \sum_{i=1}^r h_i(\delta(t))\mathbf{C}_w^i \quad (6.41)$$

The fuzzy controller consists of two parts:

The first is $\mathbf{u}_1$ and is based on the state variables of the controlled system whereas $\mathbf{u}_2$ is based on the difference between the reference input $\mathbf{r}(t)$ and system output $\mathbf{y}(t)$,

$$\mathbf{u}_1(t) = -\sum_{i=1}^r h_i(\delta(t)) \mathbf{K}_i \mathbf{x}(t) \quad (6.42)$$

$$\mathbf{u}_2(t) = \mathbf{r}(t) - \sum_{i=1}^r h_i(\delta(t)) \mathbf{C}_i \mathbf{x}(t) \quad (6.43)$$

The entire *gain-scheduled control law* is then determined by,

$$\mathbf{u} = \mathbf{u}_2(x) - \mathbf{u}_1(x) \quad (6.44)$$

$$\begin{aligned} \mathbf{u}(t) &= \mathbf{r}(t) - \sum_{i=1}^r h_i(\delta(t)) \mathbf{C}_i \mathbf{x}(t) - \sum_{i=1}^r h_i(\delta(t)) \mathbf{K}_i \mathbf{x}(t) \\ \mathbf{u}(t) &= \mathbf{r}(t) - \sum_{i=1}^r h_i(\delta(t)) [\mathbf{C}_i + \mathbf{K}_i] \mathbf{x}(t) \end{aligned} \quad (6.45)$$

$$\mathbf{u}(t) = \mathbf{r}(t) - \mathbf{K}_w(\cdot) \mathbf{x}(t) \quad (6.46)$$

with

$$\mathbf{K}_w(\cdot) = \sum_{i=1}^r h_i(\delta(t)) [\mathbf{C}_i + \mathbf{K}_i]$$

as the state feedback-gain matrix.

The closed-loop behavior of the fuzzy controller is then written as,

$$\dot{\mathbf{x}}(t) = \overline{\mathbf{A}}_w(\cdot) \mathbf{x}(t) + \overline{\mathbf{B}}_w(\cdot) \mathbf{r}(t) \quad (6.47)$$

In Equ. [6.47] $\overline{\mathbf{A}}_w(t)$ is a time-varying system matrix, satisfying the LPV in Equ. [6.7],

$$\overline{\mathbf{A}}_w(t) \in Co\{\overline{\mathbf{A}}_w^{ij} : i = 1, 2, \dots, L\}, \quad (6.48)$$

for the entire time domain the matrices $\overline{\mathbf{A}}_w^{ij}$ are formed as follows,

$$\overline{\mathbf{A}}_w^{ij} = \mathbf{A}_w^i - \mathbf{B}_w^i [\mathbf{C}_w^i + \mathbf{K}_w^i] \quad (6.49)$$

Equ. [6.49] gives the transformations $(\mathbf{A}_w^i, \mathbf{B}_w^i, \mathbf{K}_w^i)$ for the synthesis of the controller, denoted as Fuzzy Scheduler, into the standard problem $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{K}_j)$.

The gain-scheduler fuzzy control is shown in Figure [6.5], with reference input $r(t)$ applied to the system.

Once the system parameters start to change, the fuzzy controller rather select $\mathbf{K}$ from a large range computed to satisfy the H^∞ performance. At the same time, the H^∞ controller $\mathbf{K}$ is computed to satisfy the extremals of system parameter variations as expressed by Equ. [6.21].

In addition, Figure [6.6] shows the fuzzy scheduling variable $\delta(t)$ that makes the controller switch from one controller gain $\mathbf{K}_i$ to another. The figure shows the variation of two parameters, M_i and M_j , and when more parameters are required, the figure will be complicated to visualize.

From the figure, the *parametric box* represent the changes in system parameters of two dimensional, whereas the shown membership function show the extremes in parameters variation which have been obtained from the modeling procedure in chapter four.

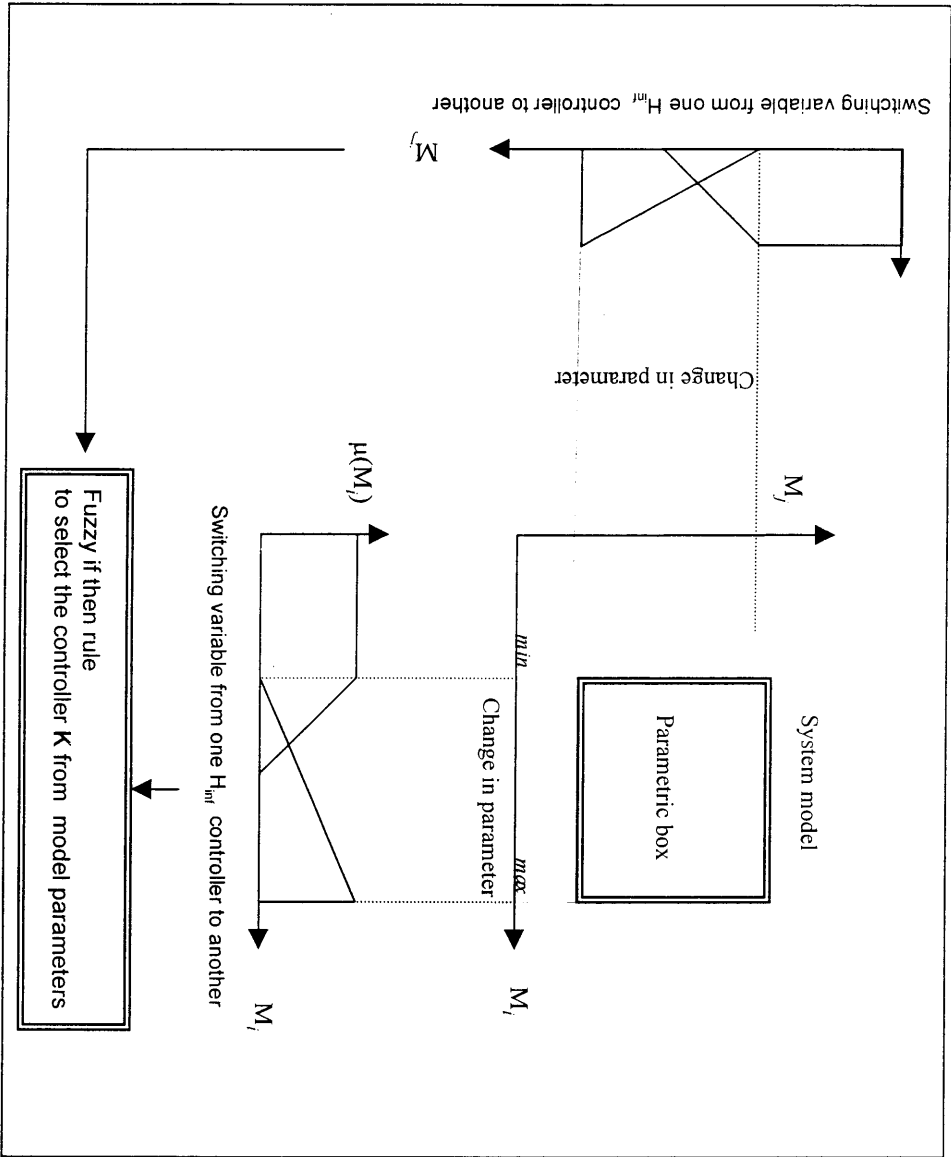


Figure [6.6] Fuzzy gain-scheduling mechanism.

6.8 Case Studies

This section presents an application of the Fuzzy H^∞ Gain-Scheduling to the two nonlinear dynamic systems already modeled in chapter four. The two dynamic systems are considered as nonlinear systems, where the parameters are assumed to be measured in real-time. Therefore, based on the linearization of the two dynamic systems around specific operating conditions, LPV representation can be developed. Hence the gain-scheduled H^∞ control presented in section 6.7 is then immediately applicable to the two problems under consideration.

[A] Fuzzy Gain-Scheduled H^∞ control of a mass-spring-damper nonlinear system:

An open-loop analysis of the mass-spring system is presented. As a result of this preliminary analysis, the motivations for using the LPV control will appear clearly.

Open Loop Analysis

With reference to the T-S fuzzy model in chapter four, the associated fuzzy linearized dynamics of the mass-spring system variables are re-described by replacing the varying parameter by variables in a state-space representation in the form of,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \nu_1 & \nu_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \zeta_1 \\ 0 \end{bmatrix} u \quad (6.50)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (6.51)$$

It should be mentioned here that; the state-space model of Equ. [6.50] (see page 101) have been obtained based on the observations of changes in parameters of the three derived fuzzy sub-models. The varying parameters ν_1, ν_2 and ζ_1 are all functions of the mass-spring system motion and are therefore available to measurement in real time.

The parametric changes (i.e. ν_1, ν_2 and ζ_1), are shown in Figure [6.7], indeed, there are abrupt changes in such parameters as functions of the mass-spring system motion that range over a large operating domain. The stability properties of the system are greatly influential.

For further validations of the final LPV controller, 18 operating points have been selected in the parameter range. These points are regularly distributed over the entire region of operation. Figure [6.8] displays the pole-zero map of the associated 18 LTI models.

Moreover, the mass-spring system can be very close to the marginally stable localities, depending on the sign of ν_1 .

The interconnection matrix $\mathbf{P}(s, \delta(t))$ of the mass-spring fuzzy model is then given by,

$$\mathbf{P}(s, \delta(t)) = \begin{bmatrix} \mathbf{A}(\delta(t)) & \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{C}_1 & D_{11} & D_{12} \\ \mathbf{C}_2 & D_{21} & D_{22} \end{bmatrix}$$

where,

$$\begin{aligned} \mathbf{A}(\delta(t)) &= \begin{bmatrix} \nu_1 & \nu_2 \\ 1 & 0 \end{bmatrix} \\ \mathbf{B}_1 &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} \zeta_1 \\ 0 \end{bmatrix}, \\ \mathbf{C}_1 &= [1 \quad 0], \quad \mathbf{C}_2 = [1 \quad 0], \\ D_{11} &= 0, \quad D_{12} = 0, \quad D_{21} = 0, \quad D_{22} = 0, \end{aligned}$$

Assuming that the augmented plant $\mathbf{P}(s, \delta(t))$ is polytopic (refer to chapter two), the LPV control problem can be described as the hull of four matrices. Each matrix corresponds to the extremal values of the two varying parameter (ν_1, ν_2) .

$$\mathbf{P}(s, \delta(t)) \in co \left\{ \begin{bmatrix} \mathbf{A}(\delta(t))_i & \mathbf{B}_{1i} & \mathbf{B}_{2i} \\ \mathbf{C}_{1i} & D_{11i} & D_{12i} \\ \mathbf{C}_{2i} & D_{21i} & D_{22i} \end{bmatrix}, \quad i = 1, 2, 3, 4 \right\}$$

with,

$$\mathbf{A}_1 = \mathbf{A}(\nu_{1 \min}, \nu_{2 \min})$$

$$\mathbf{A}_2 = \mathbf{A}(\nu_{1 \min}, \nu_{2 \max})$$

$$\mathbf{A}_3 = \mathbf{A}(\nu_{1 \max}, \nu_{2 \min})$$

$$\mathbf{A}_4 = \mathbf{A}(\nu_{1 \max}, \nu_{2 \max})$$

and, for $i = 1, 2, 3, 4$.

$$\mathbf{B}_{1i} = \mathbf{B}_1, \quad \mathbf{B}_{2i} = \mathbf{B}_2$$

$$\mathbf{C}_{1i} = \mathbf{C}_1, \quad \mathbf{C}_{2i} = \mathbf{C}_2$$

$$D_{11i} = D_{11}, \quad D_{12i} = D_{12}, \quad D_{21i} = D_{21}, \quad D_{22i} = D_{22}$$

Fuzzy Controller and H^∞ Gain-Scheduling

A two-degrees of freedom synthesis has been considered as was indicated in Figure [6.4]. Hence the design process is similar to the usual H^∞ synthesis, except that the operators for minimization are parameter-dependent. In this way, the H^∞ minimization must handle all possible trajectories in the operating domain, hence all possible models.

For the system formulation of the system introduced above and the interconnection shown in Figure [6.5], we are ready to perform the H^∞ gain-scheduled controller synthesis. The optimization algorithm converged and produced a polytopic LPV controller with a minimum γ , with structure of,

$$\mathbf{K}(s, \delta(t)) = Co\{\mathbf{K}_i(s, \delta(t)), \quad i = 1, 2, 3, 4\}$$

Fuzzy Controller Validation

The previously described controller will now be tested and analyzed. Superimposing the time responses corresponding to 50 different testing points is first considered as shown in Figure [6.9]. For each of these points, one determines its polytopic coordinate, then the corresponding system model and controller.

Furthermore, in order to perform such real simulations, one parameter trajectory has been examined. Such trajectory varies continuously in the operating domain and converges to a predefined value within one second time. Figure [6.10] shows such trajectory used in this analysis. Hence, the actual response due to such parameter trajectory is shown in Figure [6.11], where the corresponding control action is shown in Figure [6.12]. As an overall view, the system was able to converge to the required set-points with an achieved stability even under changes in the parameters. This confirms the H^∞ fuzzy gain-scheduling.

In this simulation, changes in model parameters were computed by the pre-defined trajectory, and the H^∞ controller was switching from one set of controller gain $\mathbf{K}_1$ to another set $\mathbf{K}_2$ depending on the model parameters.

It was essential in this simulation to define a well established trajectory, as shown in Figure [6.10], since any invalid parameter trajectory could results in undefined system model, i.e. outside the parameter box.

The H^∞ controller synthesis that was achieved via the LMI toolbox, was integrated to our controller simulation code that was written in MATLAB script.

In reference to Figure [6.11], fuzzy gain-scheduling was achieved with H^∞ performance (i.e. minimizing the worst case disturbance on the system input).

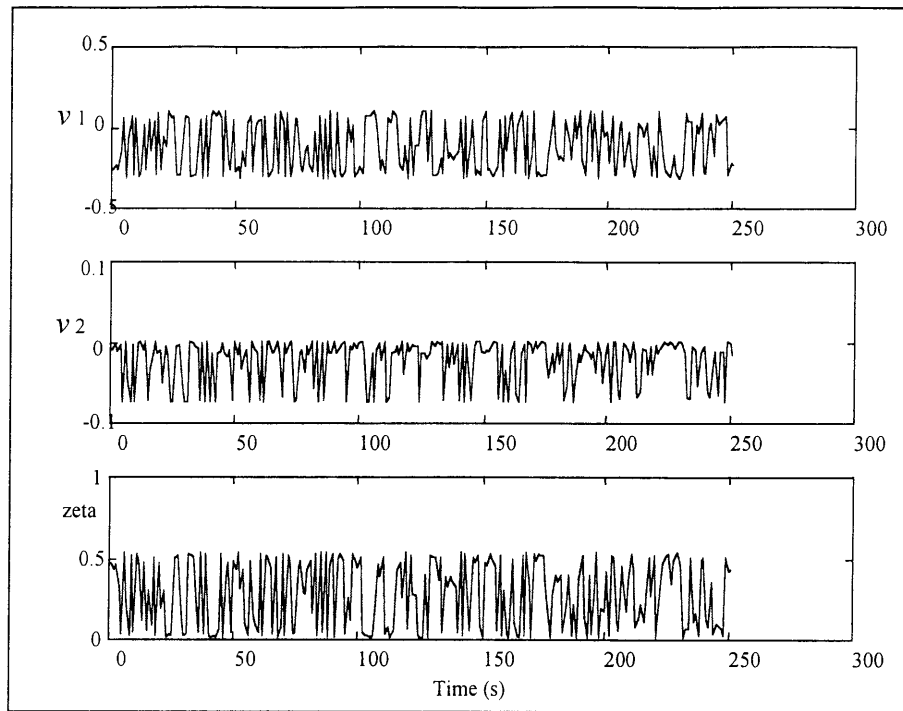


Figure [6.7] State-space parametric variation over the entire operation region for the mass-spring system.

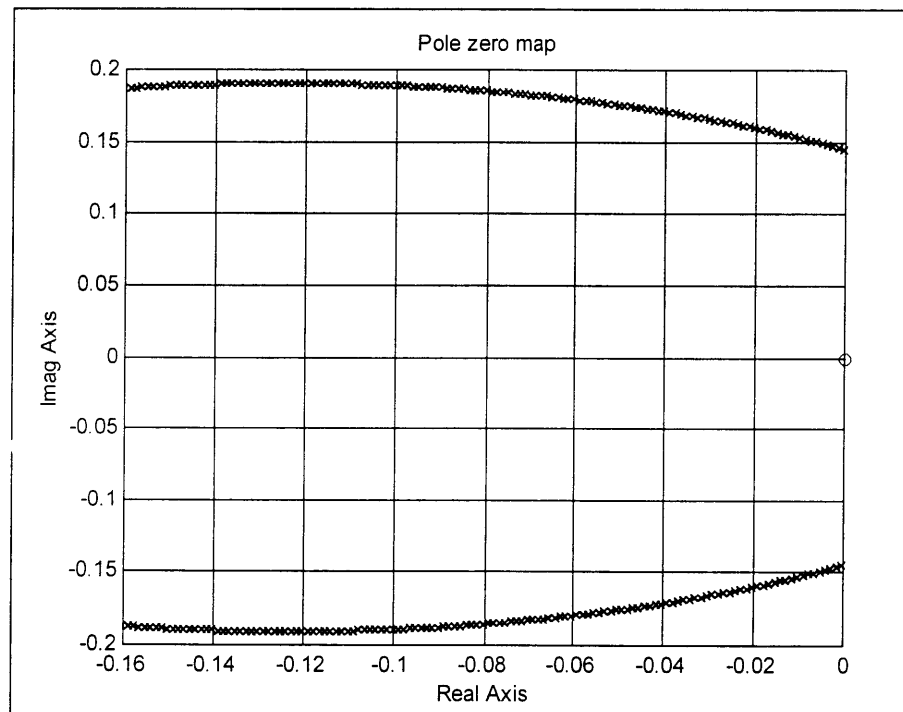


Figure [6.8] Evolution of system poles as parameters vary.
(change from critically stable to marginally stable)

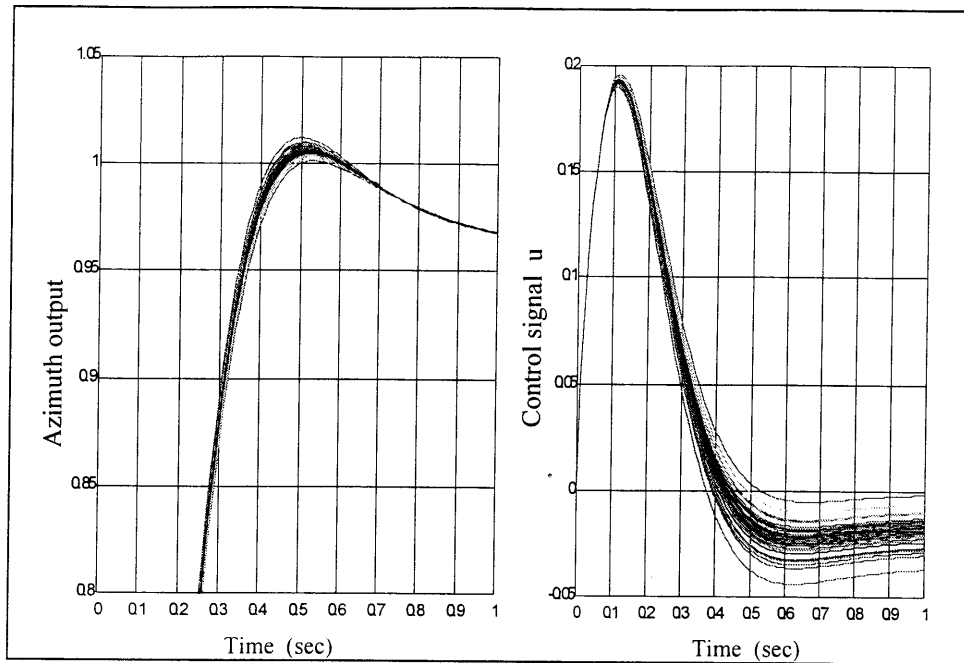


Figure [6.9] Step response of the mass-spring system subject to local simulation of 50 points, and the corresponding control action.

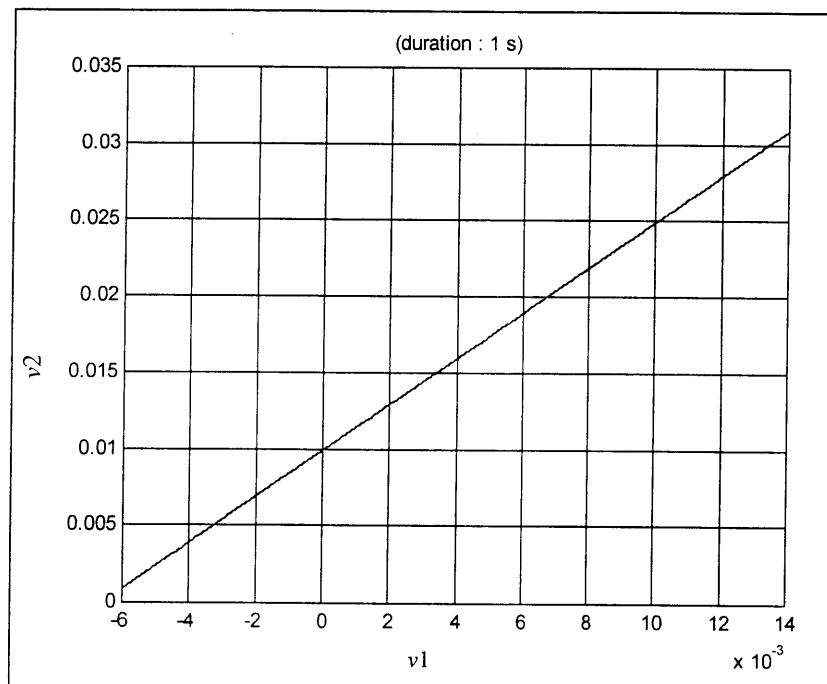


Figure [6.10] Predefined parameter trajectory

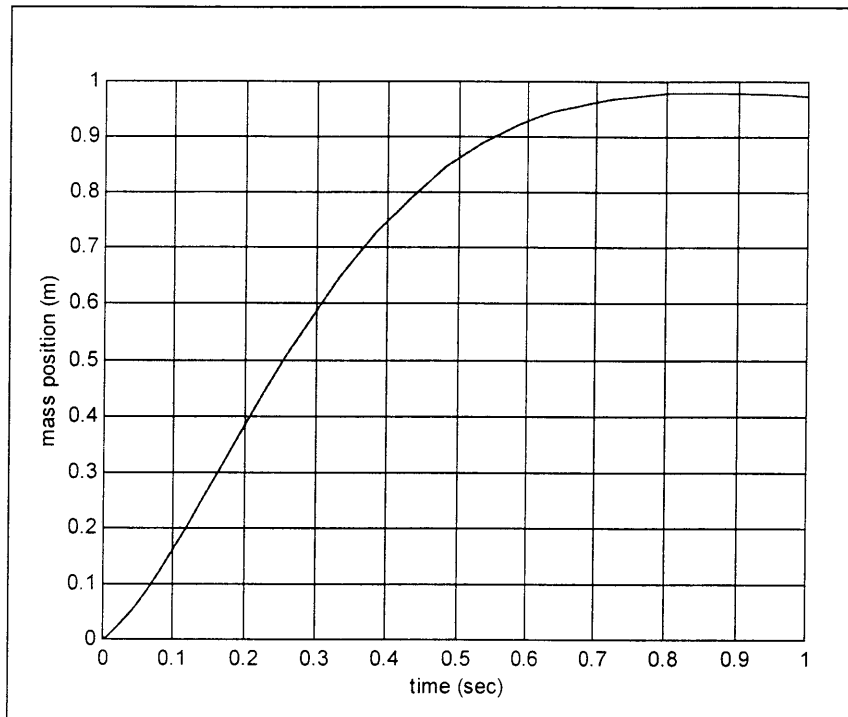


Figure [6.11] Step response of the mass position over a defined trajectory parameter.

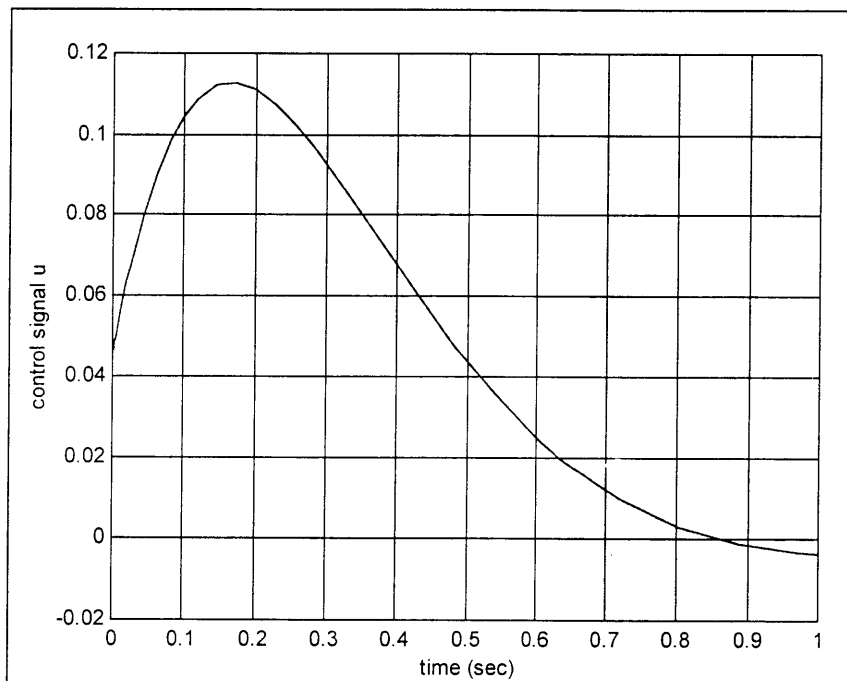


Figure [6.12] Control action for the mass-spring system.

[B] Fuzzy Gain-Scheduling H^∞ Control of an Antenna Nonlinear System

In the same way we analyzed the mass-spring system, we are going to achieve the same for the antenna system which has been introduced and modeled by a T-S fuzzy system in chapter four.

Open Loop Analysis

The associated fuzzy linearized dynamics of the antenna can be described by a state space model of the following structure,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \alpha_6 & \alpha_7 & \alpha_8 & \alpha_9 & 0 & \alpha_{10} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} \beta_1 & 0 \\ 0 & 0 \\ 0 & \beta_2 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (6.52)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (6.53)$$

It should be mention that, the given state space model of the antenna system (i.e. Eqs. [6.52-6.53] have been obtained based on the observation of the variations in the parameters of the obtained fuzzy state space sub-models over the entire operation region of operation. Hence, the varying parameters $\alpha_1, \alpha_2, \dots, \alpha_n$ are all functions of the antenna motion and are therefore available for measurement in real time.

Parametric Uncertainty

Figure [6.13a,b] and Figure [6.14a,b] show the parameters changes for the antenna system, representing variation in the state-space model. It is clear that changes in parameters are even severe for some elements. Hence this could drive the antenna system to operate in an unsatisfactory behavior.

In terms of parametric changes (i.e. $\alpha_1, \alpha_2, \dots, \alpha_n$) as was shown in Figure [6.13a,b] and Figure [6.14a,b], there are abrupt changes in such parameters as functions of the antenna

system and over the range of operation. In addition, some parameters are not showing any sever changes, hence can be ignored.

In terms of the poles and eignvalues evolution, Figure [6.15] displays the associated location of the open loop poles due to the changes in parameters for 18 operating points have been selected in the parameter range. Due to location of the poles over the right-hand side of the s-plane, this makes the construction of the controller much difficult task to be achieved.

The interconnection matrix $\mathbf{P}(s, \delta(t))$ of the antenna fuzzy model is given by,

$$\mathbf{P}(s, \delta(t)) \in co \left\{ \begin{bmatrix} \mathbf{A}(\delta(t))_i & \mathbf{B}_{1i} & \mathbf{B}_{2i} \\ \mathbf{C}_{1i} & \mathbf{D}_{11i} & \mathbf{D}_{12i} \\ \mathbf{C}_{2i} & \mathbf{D}_{21i} & \mathbf{D}_{22i} \end{bmatrix}, \quad i = 1, 2, 3, 4 \right\}$$

where,

$$\mathbf{A}(\delta(t)) = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \alpha_6 & \alpha_7 & \alpha_8 & \alpha_9 & 0 & \alpha_{10} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} \beta_1 & 0 \\ 0 & 0 \\ 0 & \beta_2 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{C}_1 = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0], \quad \mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{D}_{11} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{D}_{12} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{D}_{21} = 0, \quad \mathbf{D}_{22} = [0 \quad 0]$$

Assuming that the augmented plant $\mathbf{P}(s, \delta(t))$ is polytopic, the LPV control problem can be described as the hull of four matrices, where each matrix corresponds to the extremal values of the varying parameter $\alpha_1, \alpha_2, \dots, \alpha_n$.

$$\mathbf{P}(s, \delta(t)) \in \text{co} \left\{ \begin{bmatrix} \mathbf{A}(\delta(t))_i & \mathbf{B}_{1i} & \mathbf{B}_{2i} \\ \mathbf{C}_{1i} & \mathbf{D}_{11i} & \mathbf{D}_{12i} \\ \mathbf{C}_{2i} & \mathbf{D}_{21i} & \mathbf{D}_{22i} \end{bmatrix}, \quad i = 1, 2, 3, 4 \right\}$$

with,

$$\mathbf{A}_1 = \mathbf{A}(\alpha_{6 \min}, \alpha_{8 \min})$$

$$\mathbf{A}_2 = \mathbf{A}(\alpha_{6 \min}, \alpha_{8 \max})$$

$$\mathbf{A}_3 = \mathbf{A}(\alpha_{6 \max}, \alpha_{8 \min})$$

$$\mathbf{A}_4 = \mathbf{A}(\alpha_{6 \max}, \alpha_{8 \max})$$

and, For $i = 1, 2, 3, 4$.

$$\mathbf{B}_{1i} = \mathbf{B}_1, \quad \mathbf{B}_{2i} = \mathbf{B}_2$$

$$\mathbf{C}_{1i} = \mathbf{C}_1, \quad \mathbf{C}_{2i} = \mathbf{C}_2$$

$$\mathbf{D}_{11i} = \mathbf{D}_{11}, \quad \mathbf{D}_{12i} = \mathbf{D}_{12}, \quad \mathbf{D}_{21i} = \mathbf{D}_{21}, \quad \mathbf{D}_{22i} = \mathbf{D}_{22}$$

Fuzzy Controller and H^∞ Gain-Scheduling

For the system formulation above and the interconnection shown in Figure [6.5], we are ready to perform the H^∞ fuzzy gain-scheduled controller synthesis. The convex optimization algorithm converged and produced a polytopic LPV controller with a minimum value of γ . Its structure is described below as,

$$\mathbf{K}(s, \delta(t)) = \text{Co} \{ \mathbf{K}_i(s, \delta(t)), \quad i = 1, 2, 3, 4 \}$$

Fuzzy Controller Validation

The previously described controller will now be validated and analyzed. Superimposing the time responses corresponding to 50 different testing points is first considered as shown in Figure [6.16]. For each of these points, one determines its polytopic coordinate, hence the corresponding model and controller gain $\mathbf{K}$.

Furthermore, in order to perform such simulations, two parameters trajectory has been examined. Such trajectory varies continuously in the operating domain and converges to a

predefined value within 10 second time. Figure [6.17] shows such trajectory used in this analysis. Hence, the actual response due to such parameter trajectory is shown in Figure [6.18], where the corresponding control action is shown in Figure [6.19].

As an overall view, the system was able to converge to the required set-points and also the stability is achieved, even under changes in the parameters, hence confirming H^∞ gain-scheduling of the fuzzy system.

In reference to the shown parametric trajectory in Figure [6.17], two parameters have been selected for variation. This is due to the fact that; to clarify the concept of gain-scheduling, it will be a complicated issue to change all or most of the parameters given in Equ. [6.52]. Moreover, the parametric changes shown in Figure [6.13ab] and Figure [6.14ab], have indicated that some antenna parameters are not very much subjected to large variations. In this respect, only two parameter have been selected to follow the shown trajectory.

Regarding the controller and antenna system response, the scheduling mechanism has resulted in a good response of the system, although the two parameter (α_6, α_8), changing within the range between (-0.0512 to 0.3901) for α_6 and (0.6136 to 0.9215) for α_8 .

The fuzzy gain-scheduling has resulted in a reasonable response, with an error at the steady-state time. Such error can be reduced or even eliminated by including an integral control in the forward loop.

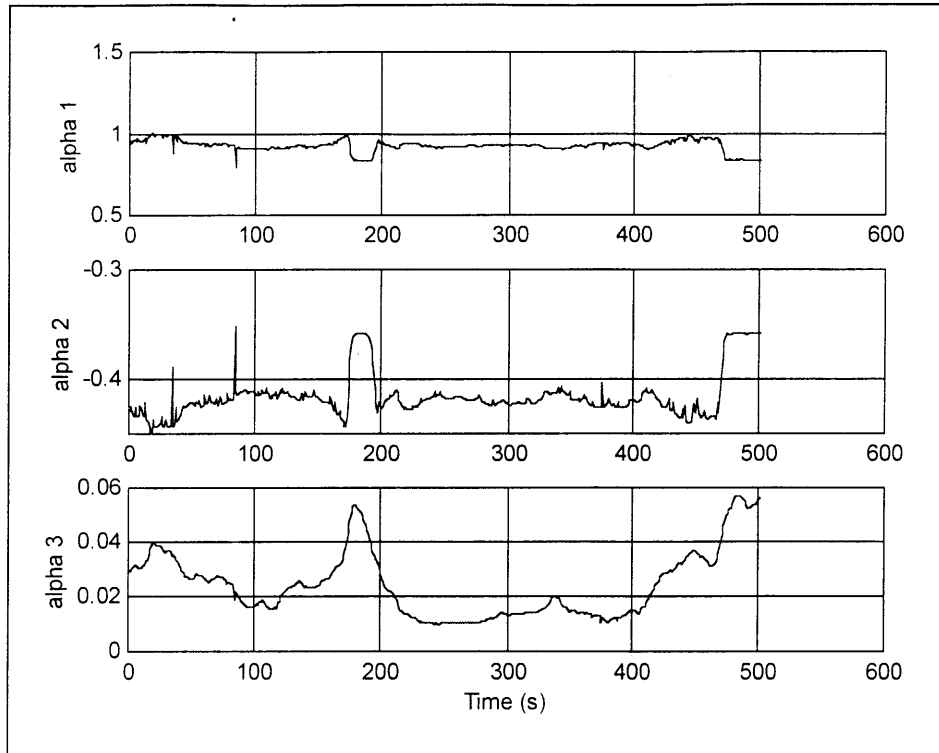


Figure [6.13a] Parameter variation of the linearized antenna system.
parameters $\alpha_1, \alpha_2, \alpha_3$

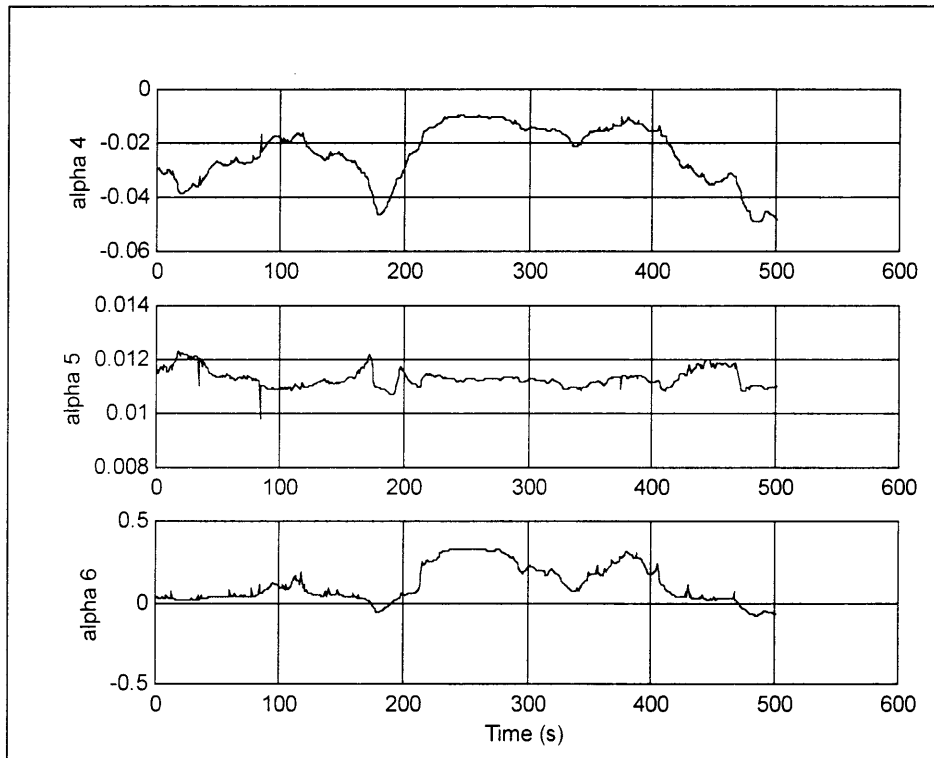


Figure [6.13b] Parameter variation of the linearized antenna system.
parameters $\alpha_4, \alpha_5, \alpha_6$

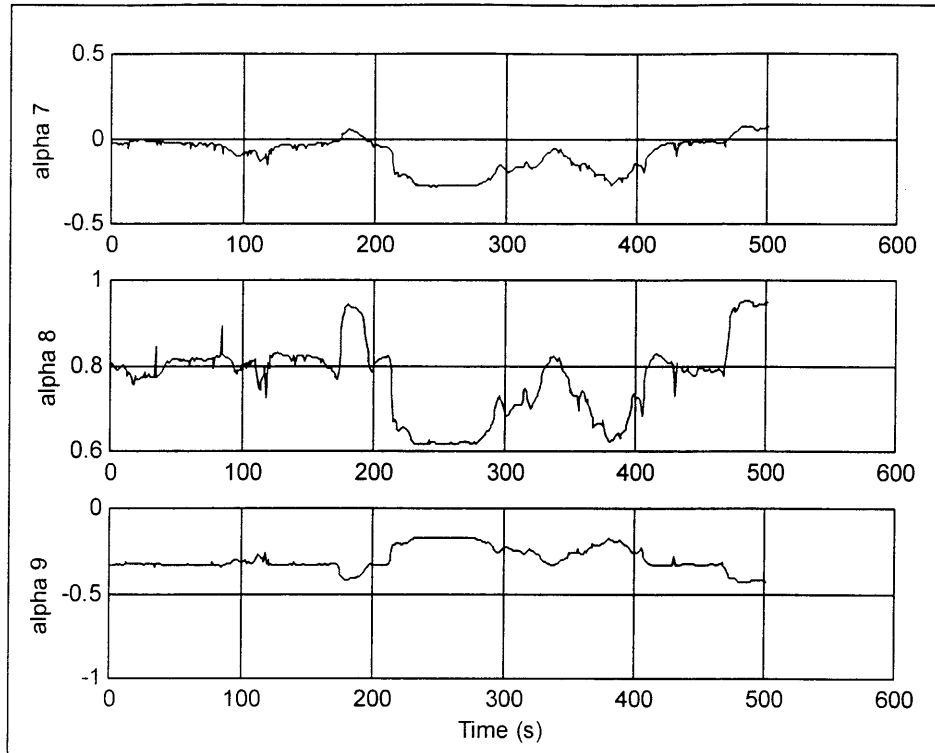


Figure [6.14a] Parameter variation of the linearized antenna system.
parameters $\alpha_7, \alpha_8, \alpha_9$

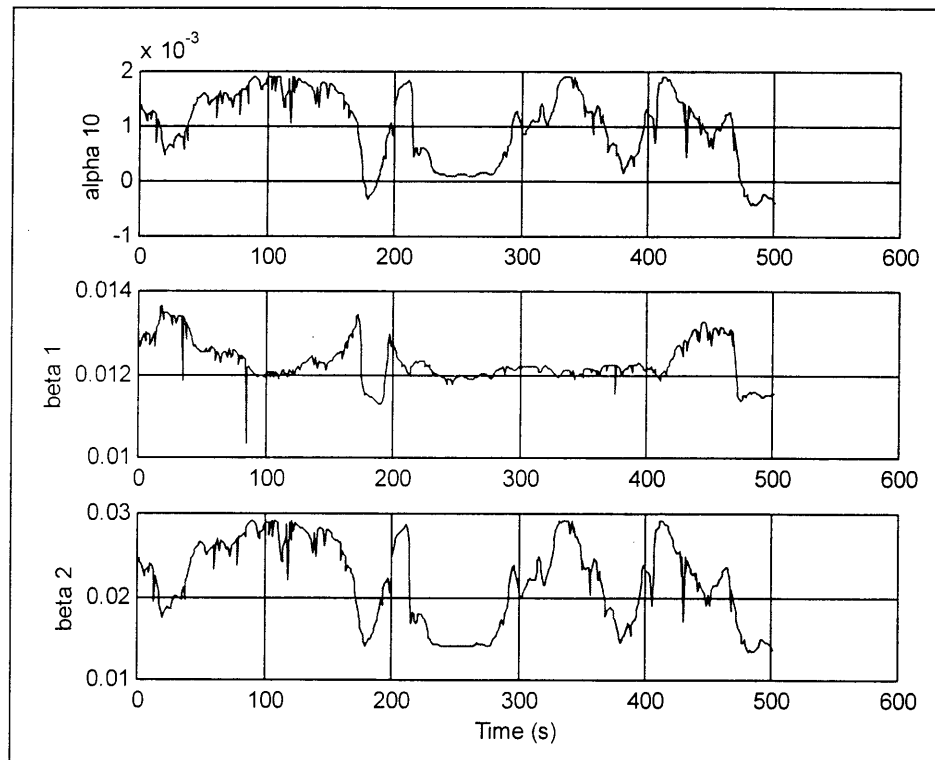


Figure [6.14b] Parameter variation of the linearized antenna system.
parameters $\alpha_{10}, \beta_1, \beta_2$

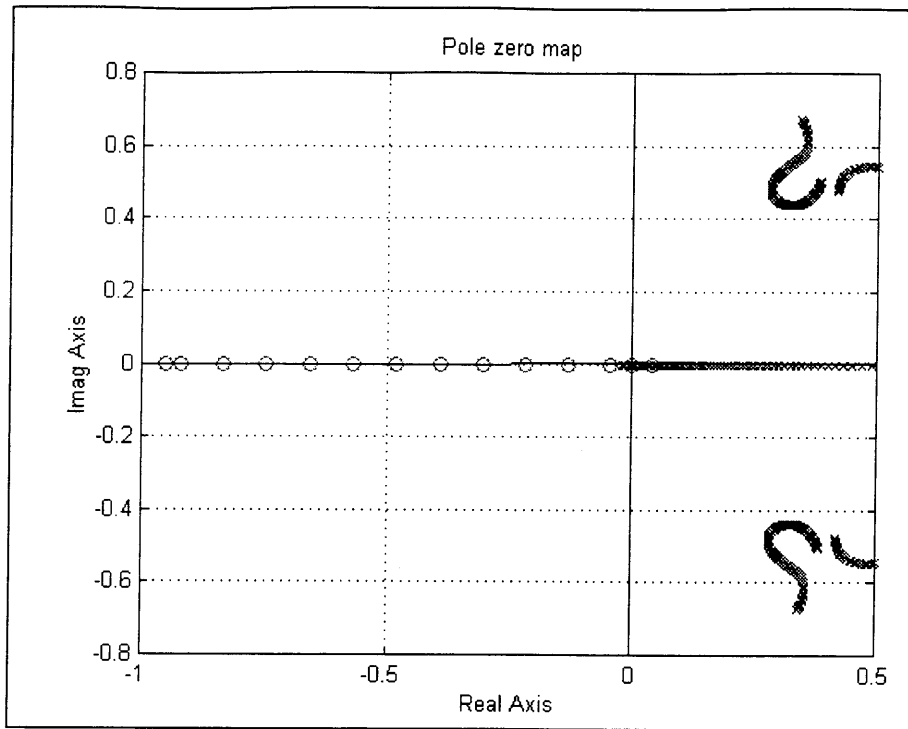


Figure [6.15] Evolution of system poles as the parameters vary (changes with the unstable region)

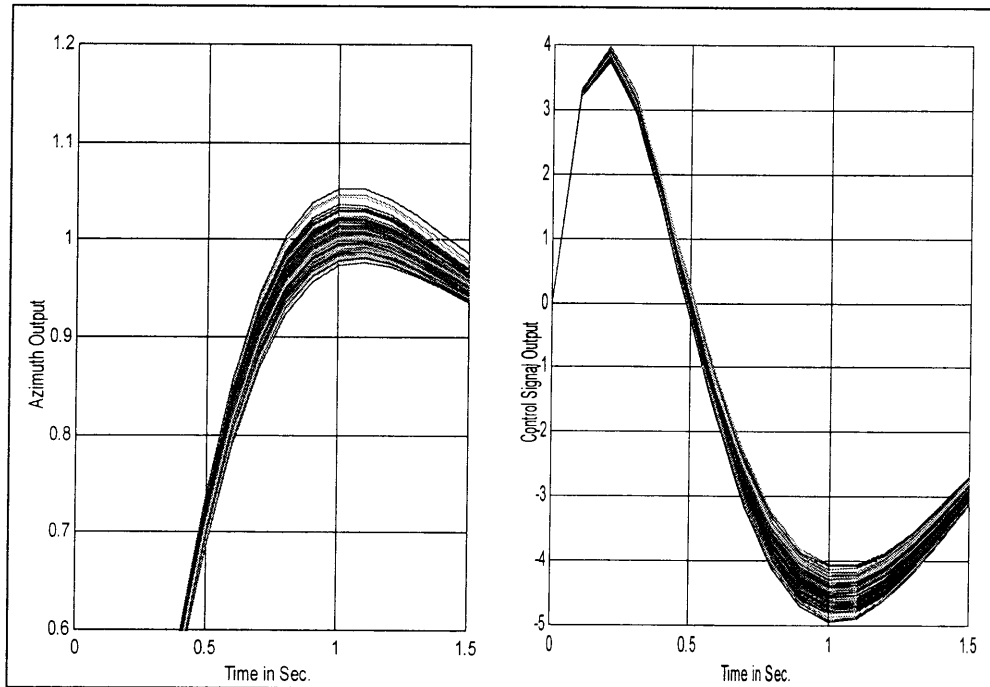


Figure [6.16] Step response of the antenna output 1 and its associate control signal system subject to local simulation of 50 points.

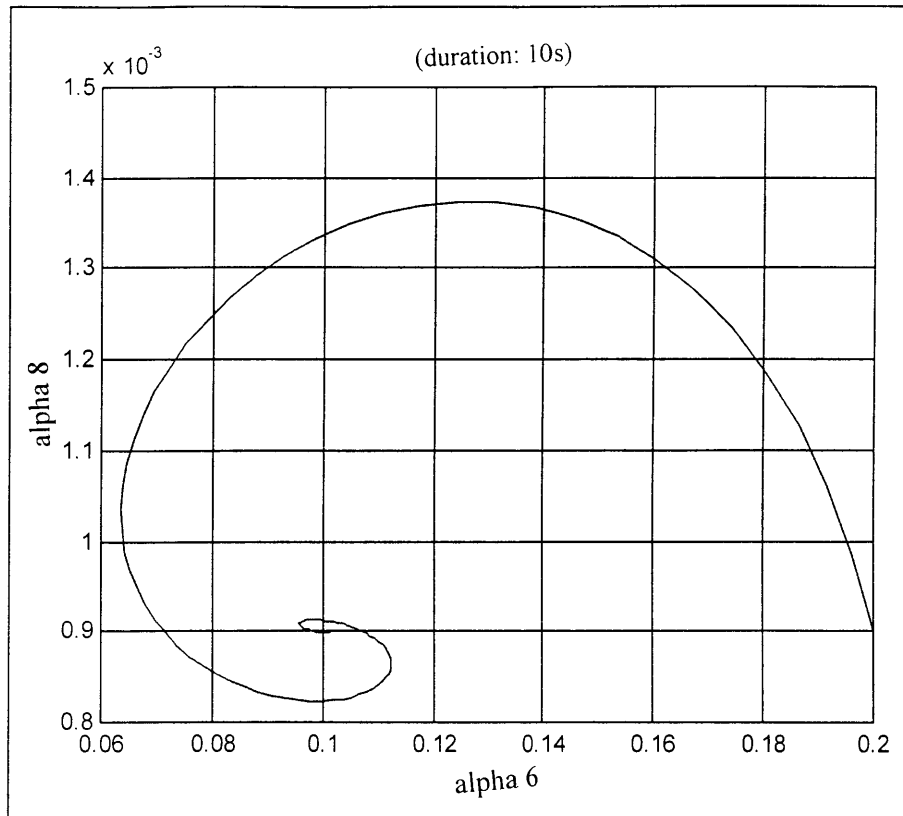


Figure [6.17] Predefined parameter trajectory for the antenna system.

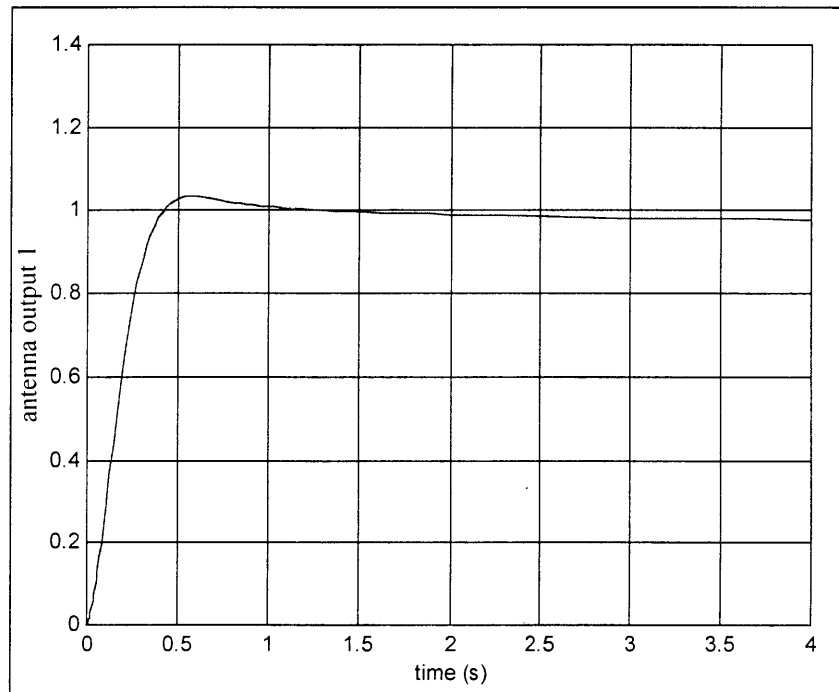


Figure [6.18] Step response of the antenna system.

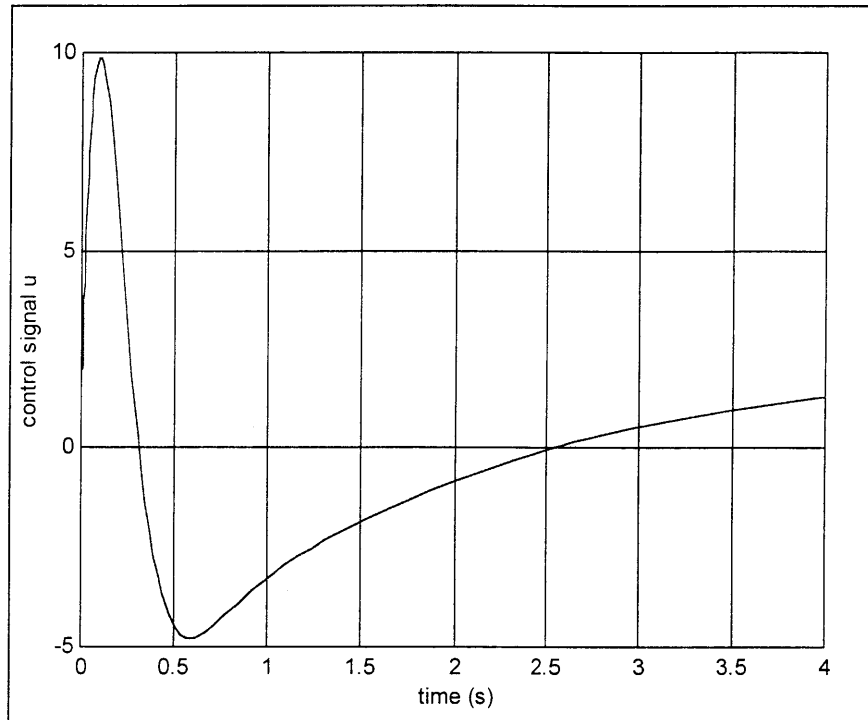


Figure [6.19] Control action for the antenna system.

6.9 Conclusions

Chapter six has been directed towards the employment of T-S fuzzy linear models in the design of H^∞ gain-scheduling fuzzy controller. The outcomes of the chapter were a number of issues regarding the design of a robust gain-scheduling fuzzy controller. The designed fuzzy controller was synthesized based on the use of the T-S fuzzy models; through the observation of the parametric variation among different sub-models. Such parameter variations are hence considered as *model uncertainty*, hence a good gain-scheduling with H^∞ characteristics to be considered.

With respect to the controller, it was constructed based on the parametric values of the system under concern, hence the fuzzy sub-models (obtained from chapter four) are used at this stage to extract the actual values of parameters.

Gain-scheduling has been achieved via constructing an artificial trajectory (such as a circle or nonlinear line trajectory), this has helped in looking at how the controller was able to adapt itself once the system under concern was exhibiting different parameters values. The synthesized H^∞ controller was able to adapt itself based on the simulated parameters trajectory.

The synthesis of H^∞ fuzzy gain-scheduling control did indeed require two main results that have been looked at in depth in this thesis. The first was the formulation of the H^∞ for LPV systems, which allows the construction of H^∞ controller in phase of parametric variations. The second was the use of the T-S fuzzy modeling technique to investigate the fuzzy linear local models in addition to the observation of how the entire global model can be considered as LPV system.

H^∞ fuzzy gain-scheduling has shown to have two main advantages; the first was the use of the linear H^∞ theories to control nonlinear plants, whereas the second was the use of fuzzy system to make the H^∞ control much transparent.

Chapter Seven:

Concluding Remarks

Concluding Remarks

This research framework has investigated a number of issues regarding the integration of *fuzzy logic* and *robust H^∞ control*. In this respect, two major issues have been looked at in this thesis that have been employed to form this class of integration. The first was the dynamic system modeling via a *neurofuzzy* system with its associated transparency characteristics for nonlinear systems modeling.

The second issue is the design of an associated H^∞ robust controller that makes the system under concern performs in accordance to a ready defined performance measure, in terms of frequency and time domain specifications.

[a] System Modeling

In reference to the first issue of the research, which is the system modeling, it was achieved via five-layered *neurofuzzy* structured with clustering algorithm. The clustering technique was employed to generate the appropriate training data centers, hence the appropriate initial center of membership functions.

Furthermore, for a closer look at the system modeling, two practical examples have been employed. The first was a SISO nonlinear mass-spring-damper system, whereas the second was a nonlinear MIMO antenna with interactions.

Takagi-Sugeno (T-S) modeling technique was implemented. Such an algorithm has the ability to sub-divide the entire nonlinear plant dynamics into linear sub-models via a ruling mechanism.

The modeling algorithm was able to model the two systems (i.e. the nonlinear mass-spring system and nonlinear antenna system) with a good degree of accuracy that reach 98% and 94% for both dynamic system respectively. The modeling methodology was found to be accurate in terms of representing the modeled systems.

However, an issue of concern was found in the literature of a difficulty to deal with systems once the number of rules is increased. This has been overcome using T-S modeling, that reduces the number of rules and consequently the number of sub-models.

The main advantage of the T-S modeling was the ability of this algorithm to sub-divide the entire operating region of the process into a number of linear sub-models which have been utilized later for H^∞ control synthesis.

Convergence to a defined tolerance while training could be a major drawback of the T-S modeling technique. Even so, preliminary knowledge of the system to be modeled is also a concern that have been found and dealt with in this thesis. This is specifically concerning the number of clusters and the number of delays in time lags.

[b] The H^∞ Robust Control Synthesis

The second main issue of the thesis was the associated robust H^∞ design for the two nonlinear systems. With respect to the design of H^∞ *nonlinear control* system, this is a very complicated mathematically formulated theory. However, via the employment of the Takagi-Sugeno modeling technique, it was possible to design linear H^∞ robust controllers for each linear sub-model. Hence, linear robust H^∞ control synthesis can be applied accordingly.

In terms of robust H^∞ design, two design methodologies have been investigated.

Regarding the first, that was H^∞ regulation problem, which was mainly concerned with designing an appropriate static feedback gains to minimize the effect of disturbance level γ on the system input. The static feedback gains were synthesized based on the linear H^∞ control theory, hence switching between different controller is achieved based on the operating conditions of the system via fuzzy rule base.

Regarding the second approach, that was based on the *gain-scheduled* H^∞ control synthesis. In the scheduled H^∞ control design, the nonlinear system was considered as uncertain dynamical system, where its parameters vary within region of limits.

[c] Gain-Scheduling H^∞ Control Synthesis

In the *gain-scheduling* H^∞ synthesis, the nonlinear systems under concern were treated as uncertain system with parametric range of variations (i.e. region of variation). Hence, scheduled H^∞ controller has been synthesis for such parameter varying systems. Knowledge of real time parameter changes is obtained via the employed fuzzy modeling technique.

From the designed H^∞ *gain-scheduling* for both dynamic systems under concern, it was found that the scheduled H^∞ controller where able to adjust controller gain parameters to follow the changes in the performance requirements for scheduled H^∞ characteristics.

Synthesizing H^∞ fuzzy *gain-scheduling* is indeed a new paradigm in the area of fuzzy control. Hence this thesis has indeed looked in depth at specific mechanisms to achieve such synthesis. Considering the global model of a nonlinear fuzzy system to schedule the H^∞ controller is one approach, however other approaches are to be investigated.

In general fuzzy system has been employed extensively to achieve a good performance of the scheduled H^∞ behavior for nonlinear system via the utilization of the linear scheduled H^∞ control theory. Fuzzy modeling was able to give a well knowledge-base models with all associated issues of transparency which is an essential for any control system design.

Appendices

Appendix A-1

Appendix A-2

[A-1] Nonlinear Spring-Mass-Damper System

The system shown in Figure [A1.1] is a SISO mass-spring-damper system. It is a nonlinear system, where it is assumed that the nonlinearity is produced by the stiffness of the spring. The system is described by the following differential equations,

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{K}{M}x_1 - \frac{B}{M}x_2 - \frac{1}{M}x_1^3 + \frac{u}{M}\end{aligned}$$

where M is the mass, K is the spring stiffness, B is the damper viscous friction coefficient and u is the force in Newton acting on the mass. The function x_1^3 is the nonlinear term with respect to the spring. y is the output of the system representing the displacement in meters of the mass object.

Representation of the system in LFT is shown in Figure [A2.1], which will be used in chapter five and six, in the design of H^∞ robust fuzzy control.

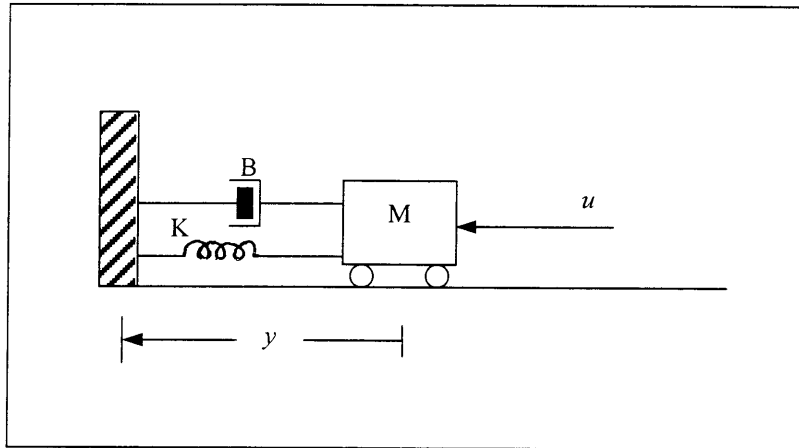


Figure [A1.1] Mass-spring-damper system.

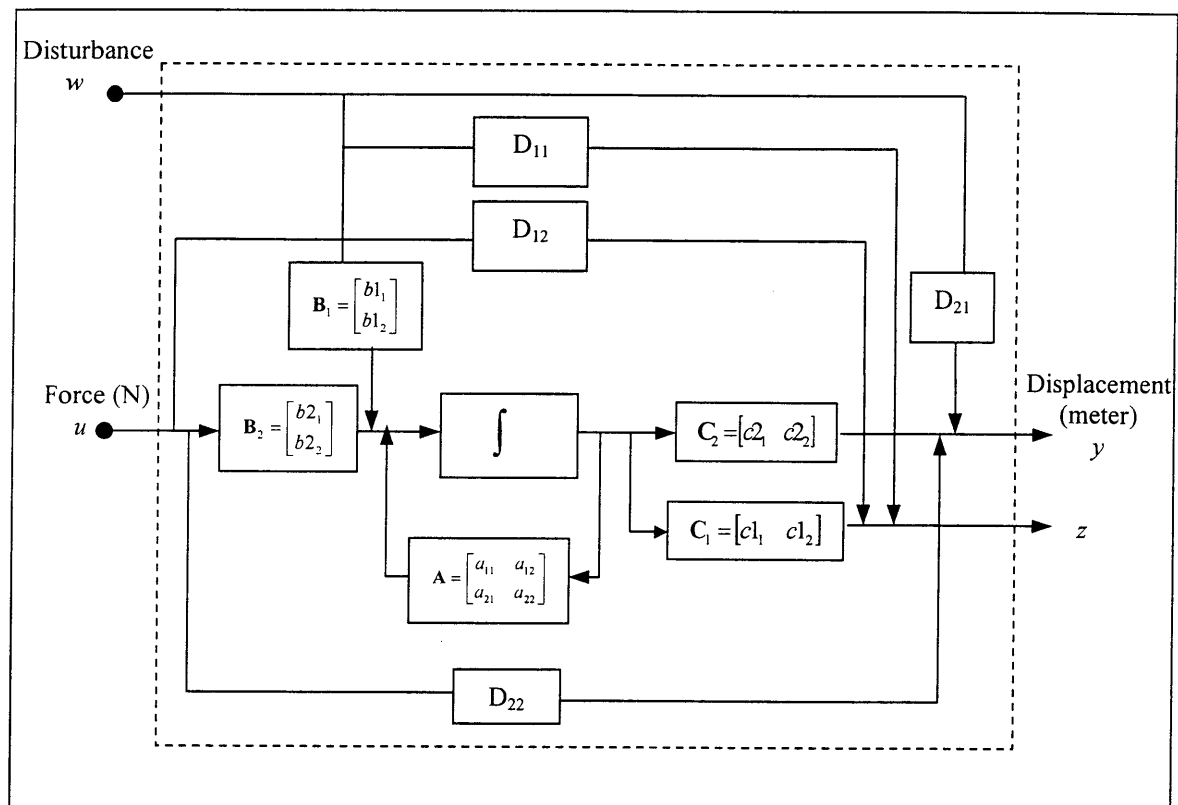
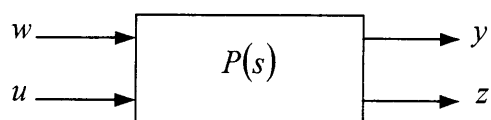


Figure [A1.2] A block diagram of LFT representation of the mass-spring system.



[A-2] Antenna System

A coupled two degree of freedom satellite dish, typical of the type used for Oceanary satellite communication systems, are presented.

The behavior of the antenna is described by the following nonlinear idealized time invariant state space equations:

$$\mathbf{z} = [\varphi \quad \dot{\varphi} \quad \psi \quad \dot{\psi}]^T$$

$$\dot{\mathbf{z}} = \begin{bmatrix} \frac{T_\varphi - b_\varphi \dot{\varphi} - \dot{\varphi} \dot{\psi} (I' - I) \sin(2\psi)}{I' \sin^2(\psi) + I \cos^2(\psi)} \\ \frac{T_\psi - b_\psi \dot{\psi} - \frac{1}{2} (I' - I) \dot{\psi}^2 \sin(2\psi)}{I} \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} e_\varphi(t) \\ e_\psi(t) \end{bmatrix}$$

where φ is the azimuth angle, ψ is the elevation angle, b_φ and b_ψ are the associated friction coefficients, and T_φ and T_ψ are the torques applied to the axes. Nominal values [52] are given in Table [A2.1]. To produce a more realistic simulation the outputs are corrupted by additive Gaussian noise, $[e_\varphi(t) \quad e_\psi(t)]^T$, representing a crude approximation to measurement noise.

	b_φ	b_ψ	I	I'
Isotropic	0.3375	0.3375	0.2 Nms ²	0.2 Nms ²
Non-isotropic	0.3375	0.3375	0.2 Nms ²	0.02 Nms ²

Table [A2.1] Antenna system parameters.

The azimuth is permitted to turn through a complete revolution, while end stops restrict the elevation to the interval $[0, \pi]$. Such antenna system can be depicted schematically in Figure [A2.1] and [A2.2].

In this antenna there are essentially two sources of nonlinearity; that produced from the end stops on the elevation and the other as a results of the non-isotropic moment of inertia tensor

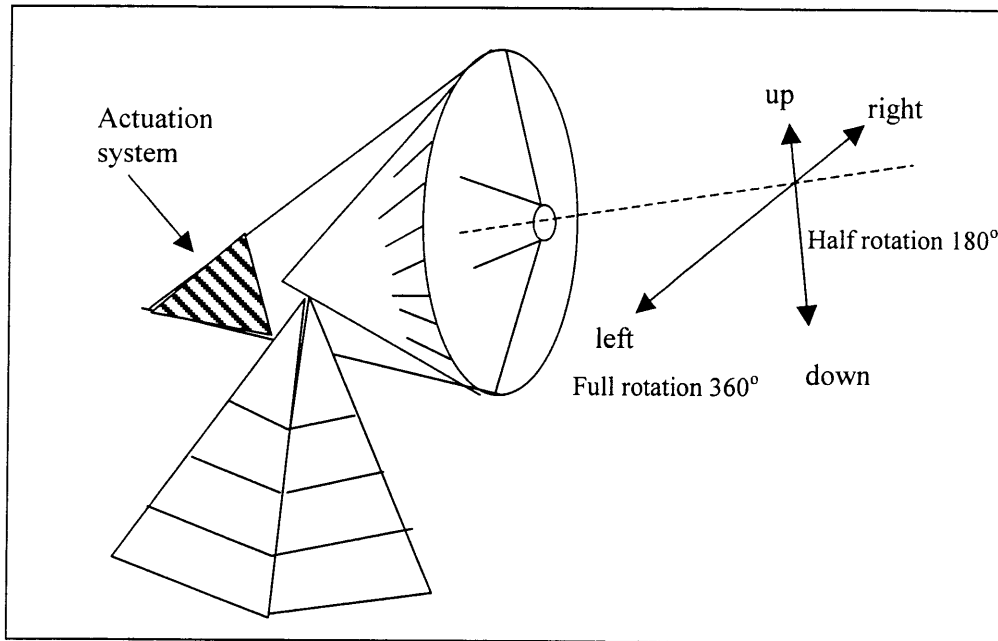


Figure [A2.1] Schematic of an antenna

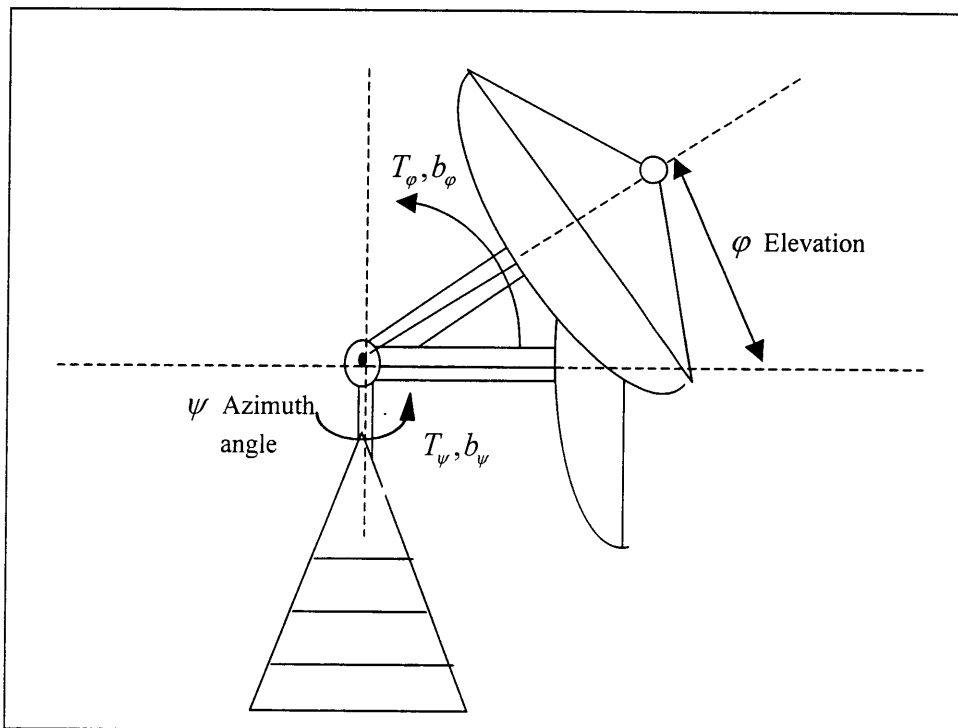


Figure [A2.2] A detailed schematic of an antenna system.

i.e. $I' \neq I$. Indeed when isotropy is present the state space equations (above) are linear. The strength of this non-linearity depends on the degree of anti-isotropy and the angular velocities of the antenna.

Neurofuzzy modeling is applied to the problem of identifying a discrete model of the dynamics of the antenna. For this task it is assumed that only measurement of the angular positions are available. However, the rate of change, or at least the direction, of the respective motions of the antenna may be very useful to the modeling process hence a regressor is constructed containing previous inputs and outputs, and previous changes in the outputs i.e.

$$\mathbf{x}(t) = [T_\varphi(t-1), \dots, T_\varphi(t-n_{T_\varphi}), T_\psi(t-1), \dots, T_\psi(t-n_{T_\psi}), (t-1) \\ \varphi(t-1), \dots, \varphi(t-n_\varphi), \psi(t-1), \dots, \psi(t-n_\psi), \Delta\varphi(t-1), \\ \dots, \Delta\varphi(t-n_{\Delta\varphi}), \Delta\psi(t-1), \dots, \Delta\psi(t-n_{\Delta\psi})]$$

where t represents discrete time intervals and Δ represents the difference in the last two time steps, $\Delta\varphi = \varphi(t) - \varphi(t-1)$. The use of this differencing can be considered as a simple method of input space projection (rotation). Assuming no a priori knowledge of the process, the maximum tap delays in the regressor are chosen to be $n_{T_\varphi} = n_{T_\psi} = n_\varphi = n_\psi = n_{\Delta\varphi} = n_{\Delta\psi} = 2$. This highlights one of the major difficulties in dynamical system modeling, even for simple low dimensional systems the input space prescribed by the size of the regressor is typically high.

Representation of the antenna in LFT is shown in Figure [A2.3], which will be used in chapter five and six, in the design of H^∞ robust fuzzy control.

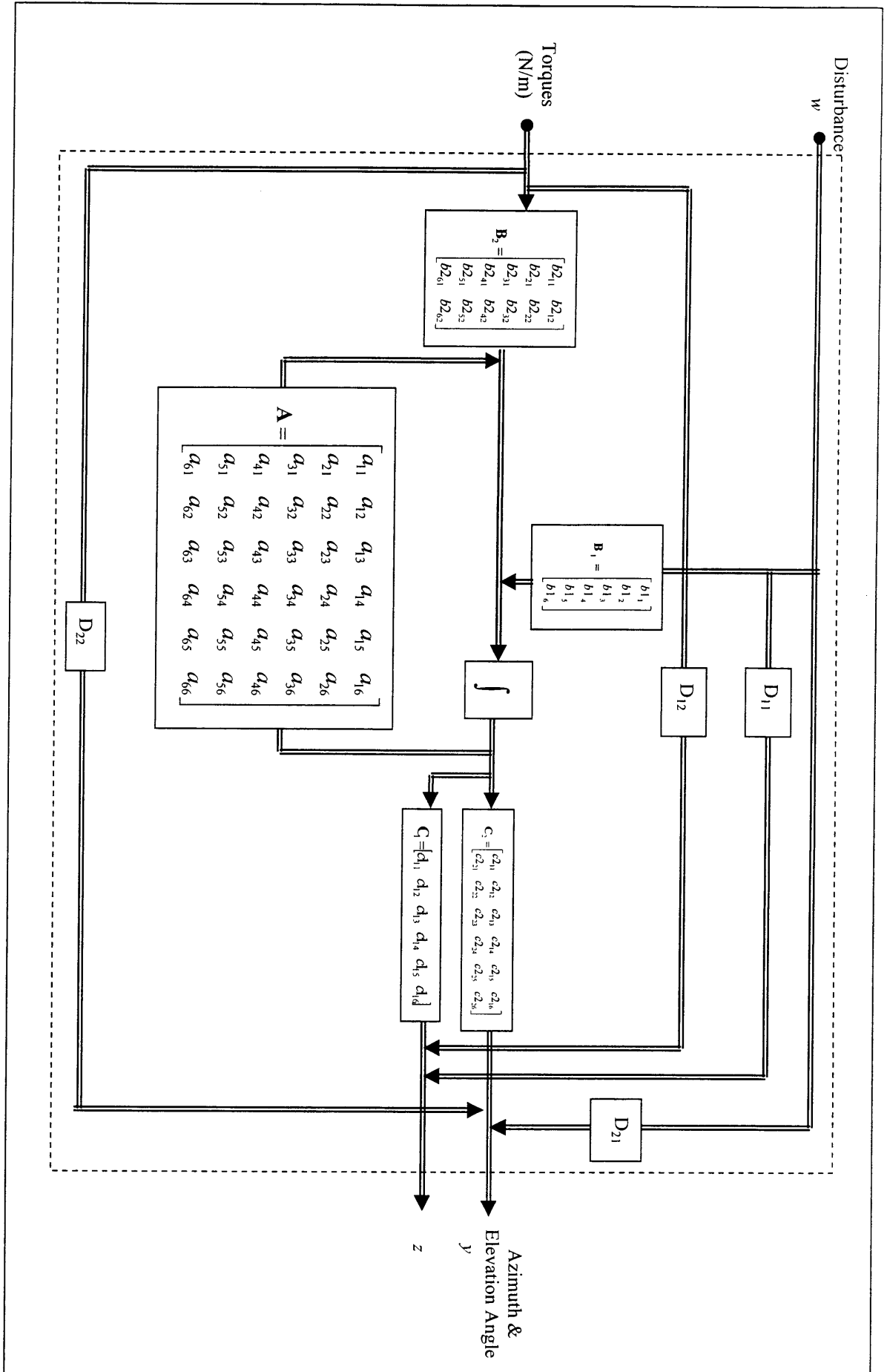


Figure [A2.3] A block diagram of LFT representation of the Antenna system.

References

- [1] Brown M. and Harris C., "Neurofuzzy Adaptive Modeling and Control", Prentice Hall, 1994.
- [2] Jang J., Sun C. and Mizutani E., "Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence", Prentice Hall, 1997.
- [3] Suykens J., Vandewalle J. and De Moor B., "Artificial Neural Networks for Modeling and Control of Non-Linear Systems", Kluwer Academic Publishers, 1996.
- [4] Zadeh L., "Fuzzy sets", Information and Control, pp. 338-353, 1965.
- [5] Lin C. and George Lee C., "A Neuro-Fuzzy Synergism to Intelligent Systems", Prentice Hall, 1996.
- [6] Mamdani E., "Applications of Fuzzy Algorithms for Simple Dynamic Plant", Proc. IEE, 121, No. 12, pp. 1585-1588, 1974.
- [7] Korba P., Babuska R., Vebruggen H. and Frank P., "Fuzzy Gain Scheduling: Controller and Observer Design by Means Of the Lyapunov Method and Convex optimization Techniques", to be appear in IEEE Trans. On Fuzzy Systems.
- [8] Mamdani E. and Assilian S., "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", International Journal of Man-Machine Studies, pp. 1-13, 1975.
- [9] Takagi T. and Sugeno M., "Fuzzy Identification of Systems and its Application to Modeling and Control", IEEE Trans. Systems, Man and Cybernetics 15(1), pp. 116-132, 1985.
- [10] Wang H., Li J., Niemann D. and Tanaka K., "T-S Fuzzy Model with Linear Rule Consequence and PDC Controller: A Universal Framework for Nonlinear Control Systems", Proc. 9th IEEE Int. Conf. on Fuzzy Systems, 2000.
- [11] Tanaka K. and Sugeno M., "Stability Analysis and Design of Fuzzy Control Systems", Fuzzy Sets & systems, vol. 16, No.4, pp.277-240, 1992.
- [12] Joh J., Chen Y. and Langari R., "On the Stability Issues of Linear Takagi-Sugeno Fuzzy Models", IEEE Transactions on Fuzzy Systems, vol.6, No.3, pp.402-410, August 1998.
- [13] Tanaka K., Ikeda T. and Wang H., "Robust Stabilization of a Class of Uncertain Nonlinear Systems via Fuzzy Control: Quadratic Stabilizability, H^∞ Control Theory, and Linear Matrix Inequalities", IEEE Transactions on Fuzzy Systems, vol.4, No.1, pp.1-13, February 1998.
- [14] Chen B., Tseng C. and Uang H., "Robustness Design of Nonlinear Dynamic Systems via Fuzzy Linear Control", IEEE Transactions on Fuzzy Systems, vol. 7, No.5, pp.571-585, October 1999.
- [15] Tanaka K. and Sugeno M., "Stability Analysis and Design of Fuzzy Control Systems", Fuzzy Sets and Systems, Vol. 45, No. 2, pp. 135-156, 1992.
- [16] Lo J. and Chen Y., "Stability Issues on Takagi-Sugeno Fuzzy Model, Parametric Approach," IEEE Transaction of Fuzzy Systems, vol. 7, No. 5, pp. 597-607, October 1999.

- [17] Tanaka K. and Sano M., "Trajectory Stabilization of a Model Car via Fuzzy Control ", Fuzzy Sets and Systems, Vol. 70, pp. 155-170, 1995.
- [18] Sugeno M., "On the Stability of Fuzzy Systems Expresses by Fuzzy Rules with Singleton Consequents," IEEE Transaction of Fuzzy Systems, vol. 7, No. 2, pp. 201-224, April 1999.
- [19] Niemann D., Li J., Wang H. and Tanaka K., "Parallel Distributed Compensation for Takagi-Sugeno Fuzzy Models: New Stability Conditions and Dynamic Feedback Designs", IFAC 1999 (Beijing), pp.207-212, July 1999.
- [20] Wang H., Tanaka K. and Griffin M., "An Approach to Fuzzy Control of Nonlinear Systems: Stability and Design Issues," IEEE Transaction of Fuzzy Systems, vol. 4, pp. 14-23, February 1996.
- [21] Li J., Niemann D., Wang H., Tanaka K., "Parallel Distributed Compensation for Takagi-Sugeno Fuzzy Models: Multiobjective Controller Design", Proc. American Control Conference, (San Diego), pp.1832-1836, 1999.
- [22] Li J., Niemann D., Wang H., Tanaka K., "Multiobjective Dynamic Feedback Control of Takagi-Sugeno Model via LMIs", Joint Conf. of Information Science, (North Carolina), pp 159-162, 1998.
- [23] Li J., Niemann D., Wang H. and Tanaka K., "Dynamic Parallel Distributed Compensation for Takagi-Sugeno Fuzzy Systems: An LMI approach", Proce. American Control Conf. (San Diego), pp.1832-1836, 1999.
- [24] Wang H., Niemann D., and Tanaka K., "T-S Fuzzy Model with Linear Rule Consequence and PDC Controller: A Universal Framework for Nonlinear Control Systems", Proc. 9th IEEE Int. Conf. on Fuzzy Systems, 2000.
- [25] Isidori A. and Asolfi A., "Disturbance Attenuation and H^∞ Control via Measurement Feedback in Nonlinear Systems," IEEE Trans. Automatic control, vol. 37, pp.1283-1293, September 1992.
- [26] Chen B., " H^∞ Tracking Design of Uncertain Nonlinear SISO Systems: Adaptive Fuzzy Approach", IEEE Transactions on Fuzzy Systems, vol. 4, No.1, pp.32-43, February 1996.
- [27] Korba P., Werner H., and Frank P. "LMI-Based Fuzzy Gain-Scheduling for the TORA Nonlinear Benchmark Control Problem", In J. Maciejowski, editor, Control 2000, Cambridge, UK, 4-7 September: 2000.
- [28] Li J., Wang H. and Bushnell L., "On the Relationship Between LMIs and AREs: Applications to Absolute Stability Criteria, Robustness Analysis and Optimal Control", Submitted to Proc. 39th CDC, Sydney, Australia, 2000.
- [29] Doyle J., "Lecture Notes on Advances in Multivariable Control", ONR/Honeywell Workshop, Minneapolis, October, 1984.
- [30] Zhou K., Doyle J. and Glover K., "Robust and Optimal Control", Prentice Hall, 1996.
- [31] Zhou K. and Doyle J., "Essentials of Robust Control", Prentice Hall, 1998.

- [32] Gahinet P., Nemirovski A., Laub A. and Chilali M., "Linear Matrix Inequalities Toolbox", The Math works, 1995.
- [33] Chilali M. and Gahinet P., " H^∞ Design with Pole Placement Constraints: an LMI Approach", Proc. Conf. Dec. Contr., pp. 553-558, 1994.
- [34] Zadeh L., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", IEEE Trans. System Man and Cybernetics, pp.28-44, 1973.
- [35] Wang L., "Adaptive Fuzzy Systems and Control, Design and Stability Analysis", Prentice Hall, 1994.
- [36] Babuska R. and Verbruggen H., "Multiple Model Approaches to Modeling and Control", Taylor and Francis, pp. 75-100, 1997.
- [37] Gustafson D. and Kessel W., "Fuzzy Clustering with a Fuzzy Covariance Matrix", Proc. IEEE CDC, San Diego, pp. 761-766, 1979.
- [38] Babuska R. and Verbruggen H., "Identification of Composite Linear Models via Fuzzy clustering", Proc. European Control Conference, Italy, pp. 1207-1212, 1995.
- [39] Ljung L., "System Identification Theory for the User", Prentice Hall, 1999.
- [40] Babuska R., "Fuzzy Modeling for Control", Kluwer Academic Publishers, Boston, USA, 1998.
- [41] Balas G. and Doyle J., " μ -Analysis and Synthesis Toolbox", The Math works, 1995.
- [42] Tani T., Murakoshi S. and Umano M., "Neuro-Fuzzy Hybrid Control System of Tank Level in Petroleum Plant", IEEE Transactions on Fuzzy Systems, vol.4, No.3, pp.360-368, August 1996.
- [43] Yi S. and Chung M., "A Robust Fuzzy Logic Controller for Robot Manipulators with Uncertainties", IEEE Transactions on Systems, Man and Cybernetics, vol.27, No.4, pp.706-713, August 1997.
- [44] Jamshidi M., "Large-Scale Systems: Modeling, Control and Fuzzy Logic", Prentice Hall, 1996.
- [45] Wang L., "A Course in Fuzzy Systems and Control", Prentice Hall, 1997.
- [46] Babuska R., "Fuzzy Modeling and Identification Toolbox, for use with Matlab", Delft University of Technology, 1998.
- [47] Gulley N. and Jang J., "Fuzzy Logic Toolbox, for use with Matlab", Mathworks, 1995.
- [48] Babuska R., "Fuzzy Systems, Modeling and Identification", Technical Report, Delft University of Technology.
- [49] Jantzen J., "Neurofuzzy Modeling", Technical Report, Technical University of Denmark, 1998.
- [50] Jantzen J., "The Self-Organizing Fuzzy Controller", Technical Report, Technical University of Denmark, 2000.

- [51] Bontempi G. and Birattari M., “ Toolbox for Neuro-Fuzzy identification and data Analysis, for use with Matlab”, 1999.
- [52] Bossley K., “Neurofuzzy Modeling Approaches in System Identification”, Ph.D. Thesis, University of Southampton, 1997.
- [53] Korba P. and Frank P., “An Applied Optimization-based Gain-Scheduled Fuzzy Control”, Proc. of American Control conference, Chicago, Illinois, June 2000.
- [54] Gahint P. and Apkarian P., “A Linear Matrix Inequality Approach to H^∞ Control”, Inter. Journal of Robust and Nonlinear Control, pp. 421-448, 1994.
- [55] Apkarian P. and Biannic J., “Gain-Scheduled H^∞ Control of a Missile via Linear Matrix Inequalities”, Journal of Guidance Control and Dynamic vol. 18, No. 3, pp.532-538.