UNIVERSITY OF SOUTHAMPTON

Local Modelling and Control of Nonlinear Systems

by

Ming Feng

A thesis submitted for the degree of
Doctor of Philosophy

in the
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

August 2000

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

Faculty of Engineering and Applied Science

Department of Electronics and Computer Science

Doctor of Philosophy

Local Modelling and Control of Nonlinear Systems

by Ming Feng

Whilst nonlinear system modelling, analysis and control are fundamentally important to a wide range of industries, they are difficult in practice due to nonlinearities and lack of precise knowledge of the systems, and therefore lack of developed theoretical and instrumental techniques. Among various efforts trying to overcome the difficulties, local schemes play an important role. Local methods are promising because: 1. Naturally any complex nonlinear system exhibits relatively simple behaviour in local areas, and 2. by obtaining simple local models for nonlinear systems, the maturely developed classical techniques such as linear theory can be employed to solve nonlinear problems.

However, currently the proposed local techniques using such as fuzzy system and neural networks are still suffering from *the curse of dimensionality*, the huge computing load in interpolation areas, and the problem of being unable to provide efficient control strategies for various nonlinear systems in practice. As an attempt to overcome some of the problems, this thesis is devoted to the development of methods for local modelling, control and stability analysis. The work of this thesis can be summarized as: 1. *Local modelling:* a new fuzzy modelling algorithm and an optimal piecewise locally linear modelling algorithm are developed. The methods are able to derive local models from experimental data of nonlinear systems and avoid the curse of dimensionality. 2. *Local Lyapunov stability:* new conditions of Lyapunov stability of local systems are derived. The conditions incorporate the input membership or interpolation function characteristics and consequently only one local Lyapunov function even in an interpolation region needs to be searched. This both relaxes the stability conditions and reduces the computation load in solving the stability problems. 3. *Controller design:* Following the modelling and stability results obtained, this thesis has formulated and solved the problem of robust feedback stabilization for a broad class of fuzzy systems. The results have two advantages compared with other methods: 1. It is capable of handling modelling error and parametric uncertainty, and 2. by using the stability results derived in this thesis, the design solutions are minimally less conservative and the design process is easy to perform as a problem of solving linear matrix inequalities. Also, the control problem of a class of nonlinear systems whose parameters are unknown nonlinear functions of the measurable operating point is solved for the most general case of an MIMO system whose operating point is completely dependent on the system states.

Some open problems for further research are discussed at the end of this thesis.

Acknowledgement

I would like to express my sincere gratitude to my supervisor Professor C. J. Harris, who has guided me through my PhD studies. Besides giving me supervision and many fruitful discussion, he has always been enthusiastic and encouraging, and have given me a lot of freedom in conducting my research. Thanks are also due to Dr. M. Brown for his invaluable help in research discussions and providing resources. I am also grateful to Professor E. Rogers and Dr C. H. Shadle for their helpful suggestions.

I would like to thank Dr X. Hong. Part of this thesis is based on a paper co-authored by her. Special thanks go to Amanda and Joanne for their continued help in preparing the thesis.

Many thanks to all the members in ISIS group for making the everyday work so enjoyable.

Lastly I would like to thank my family for their continued support during the entire period of my PhD research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Dynamic system analysis and control are fundamentally important to a wide range of industries, and as the requirements for improved performance and plant efficiency increase they are also becoming instrumental in coping with increased complexity and the need to guarantee quality of real time nonlinear systems. At the same time, increased performance demands over wide operating ranges force control engineers to move from linear to nonlinear techniques. More and more often, linear techniques fall short in analysis of complex control systems. Unfortunately, there is not a mature systematic theory and methodology for dealing with nonlinear systems. The main problems of nonlinear systems that face engineers are due to nonlinearities and lack of precise knowledge of the system dynamic, to which linear modelling and control techniques are not applicable directly. Recently, a few researchers have made efforts to overcome the difficluties. A significant result is the constructive nonlinear controller design techonology presented by French et. al. (French, Szepesvari and Rogers, 2000a; French, Szepesvari and Rogers, 2000b; French, Szepesvari and Rogers, 2000c; French and Rogers, 2000a) which concerns controlling uncertain nonlinear systems via adaptive techniques by evaluating the performance of adaptive controllers, and comparing them against eg. robust designs. This has involved developing techniques which allow lower and upper bound estimates to be made of eg. LQ performance. Uniquely in adaptive control theory, they are accounting for the control effort in the cost. Also, some motheds to apply stability theory to design a class of neuro-fuzzy feedback control schemes are introduced by (French and Rogers, 2000b).

An alternative efficient solution for this difficulty is to consider the problems *locally* because, generally speaking, a multiple local modelling approach can be more efficient in capturing the real system dynamics than a single global nonlinear model, and relatively simpler local models enable classical control design and stability

analysis methods to be applicable if and only if there is some mechanism linking local behaviour such as stability to global behaviour. Aiming at this objective, this thesis addresses the modelling and control of complex, nonlinear, *a priori* unknown systems by means of techniques based on a piecewise local model structure. Methods are described in subsequent chapters for the development of local models from data, for the design of control systems which makes use of the derived models, and for the stability analysis of the resulting composite systems.

This chapter initially gives a general view of local modelling and control in section 1.1. Then in section 1.2 the contributions of this thesis are summarized. Finally, the outline of the thesis is given in section 1.3

1.1 Local methods for modelling and control

Within the local model framework, we typically have three kinds of tasks to perform:

- modelling: including decomposition of the operating space and model parametric estimation,
- controller design, and
- stability determination of the resulting composite system.

In the following we briefly review the states in the three domains.

1.1.1 Modelling

In system engineering, modelling and identification are important steps in the design of controllers, supervision systems and fault-detection systems. Modern production and manufacturing methods in industry, combined with the growing demands concerning product lifetime, quality, flexibility in production, and safety, have increased the performance requirements placed on the control systems. Production is often characterized by frequent changes in product throughout, product mix, operating point and operating conditions. To satisfy the increasingly tight quality requirements, control systems must guarantee high performance over a wide range of operating conditions. Under these conditions, process modelling often becomes a major bottleneck for the application of advanced model-based techniques. Many systems are not amenable to conventional modelling approaches due to the lack of precise, formal knowledge about the system, strong nonlinear behaviour, a high degree of uncertainty, time varying characteristics,

etc. Examples of such systems can be found not only in the process industry, flexible manufacturing, aerospace engineering, chemical engineering, but also in ecological, social or financial domains.

A natural way to overcome the difficulties raised above is to use local modelling. Often, in a large range of complex systems, the system behaviour varies according to some kind of operating point. If the operating point space is split into a number of smaller areas, it is reasonable to expect the system behaviour to be much simpler, for example, allowing linear approximation in the local areas to a satisfactory accuracy. The overall model then can be obtained by combining the local models by means of interpolation.

A local modelling strategy needs to solve two problems, ie.,

1. how to minimally partition the operating point space, and
2. how to identify the consequent local models which includes determination of the model structure and estimating the model parameters.

Because it is always supposed that the system behaviour in local areas is relatively simple, the second problem can be solved by employing conventional mature control theories. Therefore, the research here focuses on the partitioning strategy problem. Currently, there are two kinds of partitioning methodologies. One is clustering which originated from fuzzy modelling, and the other is to recursively orthogonally partition each axis of the operating point space to a satisfactory level.

A major drawback of orthogonal partitioning is the *curse of dimensionality* (Brown and Harris, 1994). It is easy to see that, if the dimension of the operating point space is high, the number of local areas grows rapidly whilst the partitioning increases. Clustering avoids the curse of dimensionality, but usually the number of local areas needs to be known *a priori*, which obviously is a prior knowledge requirement of the unknown system, limiting the use in practice to modelling purely from data observation.

1.1.2 Stability analysis

Among various stability theories, Lyapunov techniques are very useful in system analysis. Not only do they allow stability analysis and gain computation, but they are also useful in the solution of optimal control problems. This makes Lyapunov techniques a natural basis for the analysis of local systems. Irrespective of the precise definition that we choose to use, stability is an intuitive property that a system response does not explode, ie., does not go to infinity.

Stability analysis of dynamic systems was pioneered by Lyapunov more than one hundred years ago. There is a close relation between Lyapunov stability and notions of energy. The key idea was that if every motion of a system has the property that its energy decreases with time, the system must eventually come to rest irrespective of its initial state (Lyapunov, 1892; Lyapunov, 1992). An energy measure of the system is called a *Lyapunov function* for the system. It is a very useful tool in system theory. For some systems, physical insight may hint at the selection of appropriate energy functions. But, for the great majority of systems, the choice is much less obvious. To this day, the main obstacle in the use of Lyapunov's method for a general system is the nontrivial process of finding an appropriate Lyapunov function.

Nevertheless, the situation is much simpler for linear systems of the form

$$\dot{x} = Ax. \tag{1.1}$$

Lyapunov showed that for asymptotic stability of linear systems it is both necessary and sufficient that there exists a quadratic Lyapunov function $V(\mathbf{x}) = \mathbf{x}^T P\mathbf{x}$. The conditions that such a function be proper, and that its value decreases along all motions of the linear systems result in the well-known *Lyapunov inequalities*

$$P > 0, \qquad A^T P + PA < 0. \tag{1.2}$$

In today's terminology, we would say that these conditions are *linear matrix inequalities* in P. The inequalities admit an explicit solution. In fact, by picking an arbitrary positive definite matrix $Q = Q^T > 0$, the stability of (1.1) can be assessed from the solution P to the system of linear equalities

$$A^T P + PA = -Q.$$

The system is *asymptotically stable* if and only if P is positive definite.

In the light of the above observation, a natural idea of solving the stability problems for nonlinear systems is to decompose the global system into a number of locally linear subsystems and then consider the stability problem of subsystems locally. If a nonlinear system can be decomposed into a set of locally linear subsystems, then Lyapunov functions for the subsystems can be found easily in the local area, and a global Lyapunov function, which is termed as a *piecewise Lyapunov function* in this thesis, can be formed from the local Lyapunov functions via some interpolation procedure. There are a number of different researches toward this general objective (Johansson, 1999; Branicky, 1995; Hassibi and Boyd, 1998). The key point is to develop appropriate methods to cope with the interpolation problem, about which there are two kinds of

techniques developed to date: One is making the local Lyapunov functions to be continuous in the interpolation areas, and another one is ensuring that the local Lyapunov functions are *non increasing* in the interpolation areas while allowing discontinuity. Both of the results can be reformulated as problems that require the solution of linear matrix inequalities.

A common drawback of the existing methods is that they need to consider all local models involved in a interpolation area whilst forming conditions requiring the search for local Lyapunov functions. That is to say, any single local Lyapunov function in one area has to fulfill certain conditions within all areas which have intersections with that area. Consequently, it may be required to solve a large number of linear matrix inequalities in the interpolation regions between the system submodels. In addition to the high number of linear matrix inequalities, the computation complexity and cost also increases dramatically as the input dimensionality increases. This means that the number of parameters involved in the optimization process becomes prohibitively large for large dimensional systems.

1.1.3 Controller design

The controller design techniques strongly dependent on the derived system model structure. In the local modelling structure, since the relatively simple local models are available, conventional controller design techniques can be employed. However, research is needed to develop methods of integrating the local controllers to form a global controller. There are a wide range of classes of controller design methods in this area. As representative examples, this thesis considers two particular control problems for local model structures:

1. **Feedback stabilizing control based on state space fuzzy systems**. *Fuzzy logic control* (FCL) has recently proved to be a successful control approach for complex nonlinear systems and has been suggested as an alternative approach to conventional control techniques in many case. FLC techniques represent a means of both collecting human knowledge and expertise and dealing with impression and uncertainties in the control process. Fuzzy control usually decomposes a complex system into several subsystems and uses a simple control law to emulate, say, the human control strategy in each local operating region. The global control law is then constructed by combining all the local control actions through fuzzy membership functions.

   One of the basic forms of FLC is the *feedback stabilizing controller*. The idea

is to design a state feedback controller for each local state space model and to construct the global controller from those local controllers in such a way that the global fuzzy control system is stable. Most of results to date deal with the state feedback control that requires the states of the fuzzy dynamical systems to be available or measured. In many cases, this requirement is too restrictive. Recently, there have appeared a few results of output feedback control design for fuzzy dynamical systems. However, the design methods are non-constructive, and, many iterative trials might be needed before an acceptable controller is found.

Using certain special simple structure for local models, eg., linear structure or affine structure, a feedback stabilizing problem for fuzzy systems can be transformed into a linear matrix inequality problem. Therefore, the control problem finally becomes a stability analysis problem which we described above, inheriting the same problem, ie., the high number of linear matrix inequalities in the intersection areas.

Another consideration is the robustness of the controller against modelling error and parametric uncertainty, which has been ignored in most recent research.

2. **Adaptive control based on a class of quasi-linear system models with neurofuzzy parameters**. In the design of nonlinear control systems linearisation about a set of fixed, known operating points is a natural and standard solution as it leads to a *locally* linear model for which a controller can be synthesized by classical linear control design methods. For many practical processes (e.g., aircraft gas turbines or ship dynamics), the operating point varies either as a function of independent parameters (such as mach number and altitude for an aero-gas turbine) or more usually, as a function of the system states. This phenomenon justifies the importance of a class of special local modelling structure, ie., the quasi-linear system model with operating point dependent parameters of the form (Brown and Harris, 1994; Wang, Brown and Harris, 1996)

$$\begin{aligned}\mathbf{y}(t+1) &= a_1(O_t)\mathbf{y}(t)+\cdots+a_n(O_t)\mathbf{y}(t-n+1)\\ &\quad +b_1(O_t)\mathbf{u}(t-d)+\cdots+b_m(O_t)\mathbf{u}(t-d-m+1). \qquad (1.3)\end{aligned}$$

If the system is single input single output, and the operating point is independent of the system states, an on-line adaptive controller of the 1-step ahead predictor kind has been constructed by solving a simplified Diophantine equation, which ensures tracking property. Stability of the resultant closed loop system is established (Wang, Brown and Harris, 1996).

It is clear that in practice most systems are multivariable input multivariable output (MIMO). Also, for this model structure, it is more common that the operating point dependents on the system states. For this kind of *MIMO state-dependent operating point system*, the results obtained in Wang, Brown and Harris (1996) is not applicable. Therefore, further research is needed for the general cases which obviously have something to do with strong nonlinearities.

1.2 The contributions of this thesis

This thesis is not intended to be a complete exposition of local modelling and control. Its aim is to elaborate on certain specific ideas and concepts and develop them into useful techniques, as an attempt to rise to the challenges proposed in the last section. The thesis includes the following new results:

1. Two kinds of new local modelling techniques are developed in chapter 3 for constructing data based model of a dynamical system by new methods of partitioning the data input space. The methods are able to derive a system model from data automatically and avoid the curse of dimension problem usually associated with orthogonally partition methods used conventionally in fuzzy logic, neurofuzzy algorithms and in RBF networks.

2. New Lyapunov stability conditions for local model structure are derived in chapter 5. This new approach includes a consideration of the input membership functions, via this a reduction in the number of candidate Lyapunov functions and associated linear matrix inequalities is produced. This approach significantly reduces the computational load associated with determining closed loop stability as the input dimension increases.

3. Based on the derived piecewise Lyapunov functions, new design techniques for state feedback and output feedback controllers for a class of fuzzy systems are proposed in chapter 6. The resultant controller is robust against measurement and modelling perturbations. The design conditions are non-conservative, and the design process is easy to construct by using commercially supported linear matrix inequalities software packages.

4. A neurofuzzy based scheme for modelling and control of the local model structure of (1.3) is developed in chapter 7. The model is a class of nonlinear systems with an ARMA like model (a generalized Takagi-Sugeno fuzzy model), whose

parameters are unknown nonlinear functions of the input and output variables or states of the plant. An associative memory network is used to identify each nonlinear function. The controller is a feedback linearising control law which can decouple the nonlinearity of the system. For the cases of adaptive and the fixed model parameters, detailed closed-loop stability analysis is carried out. It is shown that the consequent closed-loop system is globally stable. The main assumptions placed on the system and model for stability are minimum phase and a limit on the modeling mismatch error or uncertainty. Simulation examples are given in chapter 7 to illustrate the efficacy of the proposed approach.

1.3 Outline of the thesis

There are eight chapters in this thesis. Beside this introduction chapter, the remaining chapters are concerned with:

Chapter 2 Local modelling

After illustrating the concept of local modelling, this chapter presents an overview of the existing methods of local modelling. Two kinds of local models are discussed: Fuzzy models and quasi-linear system models of the form (1.3). Basic modelling strategies and some convergent results are introduced.

Chapter 3 Local stability

This chapter focuses on developing new local modelling algorithms. Two types of the model structures are considered. One is a kind of fuzzy models and another one is the optimal piecewise locally linear model structure. New partitioning methods are developed and local model identification is incorporated in the algorithms.

Chapter 4 Piecewise Lyapunov Stability Conditions of Fuzzy Systems

This chapter gives basic concepts and results concerning the Lyapunov stability analysis of local models. Quadratic Lyapunov functions and piecewise Lyapunov functions for both continuous time and discrete time systems are discussed. The advances of the research in this area are formulated as a number of theorems which will be used in subsequent chapters.

Chapter 5 Neurofuzzy Local Modelling

This chapter addresses the stability analysis of piecewise local systems. By incorpo-

rating the input membership function characteristics, it is shown that, under certain conditions placed on the input membership functions, we need only search for one local Lyapunov function even in the intermodel interpolation region. This both relaxes the stability conditions and reduces the computation load in solving the resultant reduced number of LMIs.

Chapter 6 Feedback stabilizing control of fuzzy systems

Following the results obtained in the preceding chapters, this chapter formulates and solves the problem of robust stabilization for a broad class of fuzzy systems. Both state feedback controller and output controller design methods are derived by seeking a piecewise Lyapunov function for the closed-loop system so that the design solutions are minimally less conservative.

Chapter 7 Adaptive Neurofuzzy Control for A Class of State Dependent Nonlinear Processes

This chapter concerns the control problems of quasi-linear systems with neurofuzzy parameters. The operating point dependent processes considered in this chapter are the most general form, ie., the multivariable input multivariable output system with state vectors as the operating points. A stable decoupling controller design method is developed. If the coefficients are *a priori* unknown, a modified recursive least square algorithm combined with feedback linearising controller is employed to design an adaptive control system. For both cases, the closed loop system stability is analysed in detail and the weight convergence is shown to be guaranteed, for practical implementation the resultant controllers can be realized as conventional controllers or as neurofuzzy controllers.

Chapter 8 Conclusions

This chapter gives the conclusions of the thesis, together with some suggestions for further research.

The content from chapter 3 to chapter 7 are all original contributions, which have been published in:

- *Advances in Neurofuzzy Algorithms for Real-time Modelling and Control*, J. Engineering Application of AI, Vol. 9, 1996, p.1-16.
- *Aspects of the Theory and Application of Intelligent Modelling, Control and Estimation*, Proc. 2nd Asian Control Conference, 1997, p.1-10.

- *Adaptive Neurofuzzy Control for A Class of State Dependent Nonlinear Processes*, International Journal of System Science, Vol 29, 1998, p.759-771.
- *A new partitioning approach for constructing fuzzy models*, Proceedings of 3rd World Multiconference on Systemics, Cybernetic and Informatics (SCI'99) and 5th International Conference on Information Systems Analysis and Synthesis (ISAS'99), Orlando, Florida, USA, July 31 - August 4, 1999.
- *Optimal Piecewise locally linear modelling*, SPIE AeroSense'99,Applications and Science of Computational Intelligence II, Orlando, USA, 1999.
- *Piecewise Lyapunov stability conditions of fuzzy systems*, Submitted to the IEEE Transactions on SMC, 1999.
- *Feedback stabilization of fuzzy systems via linear matrix inequalities*, Accepted for publication, International Journal of System Science, To appear 2000.

Chapter 2

Local modelling

2.1 Introduction

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

While most real processes are complex and nonlinear, unfortunately, the majority modelling methods are currently developed to a mature level for linear systems only (Harris, Brown, Bossley, Mills and Feng, 1996; Harris, Wu and Feng, 1997; D'Azzo and Houpi, 1995). As a result, many practical systems are not amenable to conventional modelling approaches.

To overcome such difficulties, one solution is to model a globally nonlinear system in terms of a series of local models. In fact, any model will only have a limited range of validity. This may be restricted by the modelling assumptions for a mechanistic model, or by the experimental conditions under which the data was logged for an empirical model. Global modelling is complicated, or maybe impossible because of the need to describe the interactions between a large number of phenomena that appear globally. Alternatively, local modelling may be considerably simpler because locally there may be a smaller number of phenomena that are relevant, and their interactions are less complex. Consequently, within a local modelling framework, it is possible to construct relatively simpler local models and associated controllers. For example, the local nature of model representation has computational advantages when the model is applied for model based control, and in particular adaptive control. One reason is that only a few model parameters will be relevant at a given instant. Hence, drift phenomena will be reduced in an on-line parameter estimator caused by lack of global

persistence of excitation.

Alternatively if linear models are selected as local models, the well-established linear techniques (D'Azzo and Houpi, 1995) are directly applicable to nonlinear systems, which is clearly a great advantage of local modelling.

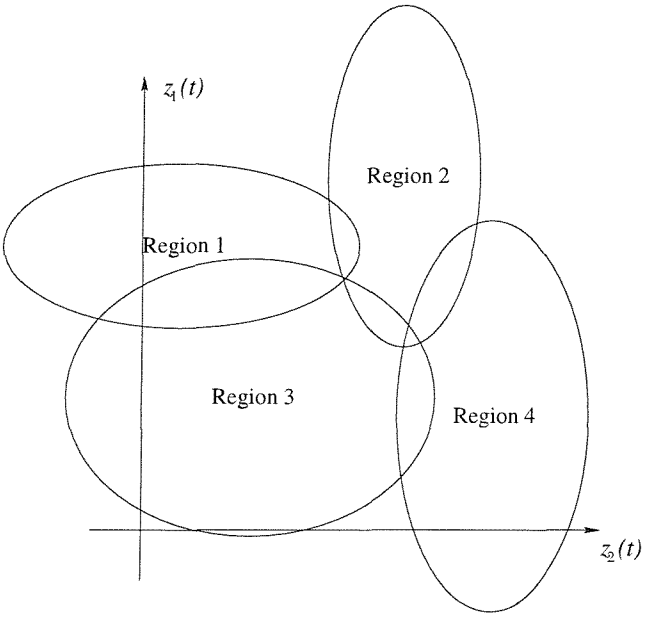

Figure 2.1: The set of two-dimensional operating points is decomposed into four regions

Local modelling framework can be conceptually illustrated as in Figure 2.1, which is adapted from Johansen (1994b). The system's full range of operation (which we refer to as the *operating point space*) is completely covered by a number of possibly overlapping *operating regions*. In each operating region the system is modeled by a *local model*, and the local models can be combined into a *global model* using an interpolation technique between each region. Design approaches differ from each other by their policies of operating point space partitioning, local model structure, and interpolation strategies. This chapter describes a number of modelling approaches using clustering or axis orthogonal partitioning, common neurofuzzy interpolation strategies. The local models can be linear, affine or nonlinear; discrete time or continuous time; input-output or state space models. In next chapter, two new modelling approaches will be introduced which have new partitioning and interpolation strategies.

Before get into details of local modelling, let us first have a look of various types nonlinear systems which will be the study objects in subsequent chapters.

2.1.1 Discrete-time nonlinear systems

Suppose a *multi-input multi-output (MIMO) nonlinear system* can be described by a discrete time series model

$$\mathbf{y}(t+1) = \mathbf{g}(\mathbf{y}(t), \cdots, \mathbf{y}(t-n+1), \mathbf{u}(t-d), \cdots, \mathbf{u}(t-d-m+1)). \tag{2.1}$$

where $\mathbf{g}(\cdot)$ is an unknown nonlinear input/output (I/O) function, $\{\mathbf{u}(t), \mathbf{y}(t)\}_{t=1}^{N}$ are measured input/output vector pairs, the integers n and m are known *a priori* or assumed system orders and d is the known time delay of the system.

Define

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{y}(t-n+1) \\ \mathbf{y}(t-n+2) \\ \cdots \\ \mathbf{y}(t) \end{bmatrix}, \qquad \mathbf{u}_t = \begin{bmatrix} \mathbf{u}(t-d-m+1) \\ \mathbf{u}(t-d-m+2) \\ \cdots \\ \mathbf{u}(t-d) \end{bmatrix},$$

and

$$\mathbf{f}(\mathbf{y}_t, \mathbf{u}_t) = \begin{bmatrix} \mathbf{y}(t-n+1) \\ \cdots \\ \mathbf{y}(t-1) \\ \mathbf{g}(\mathbf{y}(t), \cdots, \mathbf{y}(t-n+1), \mathbf{u}(t-d), \cdots, \mathbf{u}(t-d-m+1)) \end{bmatrix}.$$

Then we obtain a nonlinear discrete-time multivariable dynamical system of the form

$$\mathbf{y}_{t+1} = \mathbf{f}(\mathbf{y}_t, \mathbf{u}_t), \tag{2.2}$$

generating an output sequence $\{\mathbf{y}_t\}$. Here $\mathbf{f}(\cdot)$ is some unknown nonlinear vector function. We define $\mathbf{x}_t = [\mathbf{y}_t^T \ \mathbf{u}_t^T]^T$, so that (2.2) becomes

$$\mathbf{y}_{t+1} = \mathbf{f}(\mathbf{x}_t). \tag{2.3}$$

If $\mathbf{y}(t) \in \Re^p$, $\mathbf{u}(t) \in \Re^q$, then $\mathbf{x}_t \in \Re^{np+mq}$.

Remark: The expression of (2.3) essentially means that all MIMO systems can be considered as a surface mapping from a $(np+mq)$ dimensional space into p dimensional output space, hence the use of any neural network approximation algorithm for modelling (2.3) (or (2.1)) from observation data $\{\mathbf{x}_t, \mathbf{y}_t\}$ (Brown and Harris, 1994).

The state space counterpart of (2.1) is

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t). \end{aligned} \tag{2.4}$$

where the notations of $\mathbf{f}$ and $\mathbf{g}$ have nothing to do with those in (2.1), (2.2) and (2.3).

2.1.2 Continuous-time nonlinear systems

Many modelling and control algorithms are expressed in the continuous time domain, using measured variable which assess the state of the plant, the control signal and the desired plant's response, in order to predict the change in the plant's state (model) or to calculate the required change in control signal necessary to make the plant behave as required. A general unknown *continuous-time nonlinear dynamical system* can be described by (Brown and Harris, 1994)

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t), \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t),t). \end{aligned} \tag{2.5}$$

Given a set of input/output data $\{\mathbf{y}(t),\mathbf{u}(t)\}$, we are able to build a global nonlinear model for system (2.5). But if we do not further specify the structure of the functions f and g in (2.5), the model (2.5) is not very useful in control and estimate problems. In practice we often need models with which it is easy to determine stability, design controllers, or estimate parameters. For example, sometimes we need to consider the *affine model*,(affine with respect to input variables) which is a special case of (2.5):

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t),t)+\mathbf{g}(\mathbf{x}(t),t)\mathbf{u}(t), \quad \mathbf{x}(t)\in\Re^n, \mathbf{u}(t)\in\Re^m, \\ \mathbf{y}(t) &= C\mathbf{x}(t). \end{aligned} \tag{2.6}$$

2.1.3 Modelling and identification of nonlinear systems

Generally speaking, modelling and identification deals with the mapping of prior knowledge and empirical data to a model. While the former concerns finding a set of models that is likely to contain a model that describes the desired aspects of the system adequately, the latter needs to find a model within the model set that minimizes an objective criterion which measures the mismatch between the prediction of a given model and the empirical data. Obviously they are two stages involved in building a model for a plant. However, in this thesis, the term of modelling means the whole process of the model building, ie., given empirical data of the systems (2.1) - (2.6), find a suitable model which is able to describe the system to satisfactory accuracy to meet the application needs. There are a wide range of model structures from traditional mathematics models to intelligent model structures, among which fuzzy, neurofuzzy, and artificial neural networks (Brown and Harris, 1994) models are within the local modelling framework. As a typical representation, the fuzzy model structure is chosen to illustrate the local model idea. Also, another representive chosen in this thesis is a class of operating point dependent quasi-linear model described in (1.3).

In the remainder of the chapter, a review of the two kinds of typical local modelling methods is given. The first local model structure is fuzzy models which is reviewed in section 2.2 including its general description, and the constructing methods in section 2.3. A few useful types of Takagi-Sugeno fuzzy model are given in section 2.4. In section 2.5, the quasi-linear model structure of (1.3) is discussed. Finally, section 2.6 concludes the chapter.

2.2 Fuzzy modelling

The concepts of fuzzy-set theory can be employed in the modelling of systems in a number of ways. The most often used are *rule-based fuzzy systems* (Zadeh, 1973), *fuzzy linear regression* (Tanaka, Uejima and Asai, 1982) which generalizes conventional linear regression models by using fuzzy numbers as the model parameters, and *fuzzy models using cell structures* (Smith, Nokleby and Comer, 1994) which are a fuzzy variant of the systems based on cell-to-cell mappings (Hsu, 1980; Hsu, 1987). This thesis deals only with rule-based fuzzy systems, ie., systems where the relationships between variables are represented by a means of fuzzy if-then rules and an associated fuzzy inference mechanism. Fuzzy if-then rules take a general form:

If *antecedent proposition* **then** *consequent proposition.*

Depending on the particular form of the consequent proposition and on the structure of the rule base, three types of rule-based fuzzy models are distinguished:

1. *Linguistic fuzzy model* (Zadeh, 1973; Driankov, Hellendoorn and Reinfrank, 1993), where both the antecedent and consequent are fuzzy propositions.

2. *Fuzzy relational model* (Tong, 1979; Yi and Chung, 1993), which can be regarded as a generalization of the linguistic model, allowing one particular antecedent proposition to be associated with several different consequent propositions via a fuzzy relation.

3. *Takagi-Sugeno fuzzy model* (Takagi and Sugeno, 1985; Sugeno and Tanaka, 1991), where the consequent is a crisp function of the antecedent variables rather than a fuzzy proposition.

2.2.1 Takagi-Sugeno fuzzy model

A rule-based model suitable for the approximation of a large class of nonlinear systems was introduced by Takagi and Sugeno (1985). Concerning nonlinear model, the most

general form of a Takagi-Sugeno fuzzy rule is

$$R_i: \textbf{ If } \mathbf{z} \text{ is } A_i \textbf{ then } \mathbf{y}_i = \mathbf{f}_i(\mathbf{x}),\ i = 1,2,...L, \tag{2.7}$$

where R_i denotes the ith rule and L is the number of rules in the rule base. The first part of the rule, called the *antecedent*, is defined as a fuzzy proposition "$\mathbf{z}$ is A_i" where $\mathbf{z} \in \Re^d$ is a crisp vector called *operating point*, and A_i is a fuzzy set defined by its (multivariate) *membership function*

$$\mu_{A_i}(\mathbf{z}) : \Re^d \rightarrow [0,1].$$

The *degree of fulfillment* (*degree of truth*) $\bar{w}_i(\mathbf{z})$ of the antecedent proposition for a given value of the vector $\mathbf{z}$ is evaluated as the degree of membership of this vector into the set $A_i : \bar{w}_i(\mathbf{z}) = \mu_{A_i}(\mathbf{z})$. By means of fuzzy sets, the input space is partitioned into smaller regions, in which the modelling problem becomes more tractable.

Since the consequents in the TS model are not fuzzy sets but crisp functions, the *inference* and *defuzzification* reduced to a simple algebraic expression, similar to the fuzzy-mean defuzzification formula (Takagi and Sugeno, 1985):

$$\mathbf{y} = \frac{\sum_{i=1}^{L} \bar{w}_i(\mathbf{z})\mathbf{y}_i}{\sum_{i=1}^{L} \bar{w}_i(\mathbf{z})},$$

where $\bar{w}_i(\mathbf{z})$ is the degree of fulfillment of the antecedent of the ith rule and $\mathbf{y}_i$ is the output of the local consequent model of that rule.

Defining the normalized degree of fulfillment $w_i(\mathbf{z}) = \bar{w}_i(\mathbf{z}) / \sum_{j=1}^{L} \bar{w}_j(\mathbf{z})$, for the model (2.7) the global output can be written simply as a weighted sum

$$\mathbf{y} = \sum_{i=1}^{L} w_i(\mathbf{z})\mathbf{f}_i(\mathbf{x}). \tag{2.8}$$

This representation is appealing, since many real systems naturally change behavior smoothly as a function of the operating point $\mathbf{z}$, and the soft transition between the regions introduced by the fuzzy sets representations captures this feature in an elegant fashion. This makes TS model suitable for the approximation of a large class of nonlinear systems and therefore being considered as one of the most interesting approaches in the local modeling literature.

If the $\mathbf{f}_i(\mathbf{x})$ in (2.8) is chosen to be $A_i^T\mathbf{x} + B_i$, then we get the *affine TS model*

$$\mathbf{y} = A(\mathbf{z})\mathbf{x} + B(\mathbf{z}). \tag{2.9}$$

where $A(\mathbf{z})$ and $B(\mathbf{z})$ are convex linear combinations of the consequent parameters A_i and B_i, ie.,

$$A(\mathbf{x}) = \sum_{i=1}^{L} w_i(\mathbf{z})A_i, \quad B(\mathbf{x}) = \sum_{i=1}^{L} w_i(\mathbf{z})B_i$$

The special structure of (2.9) facilitates the analysis of affine TS models in the framework of *polytopic systems* (Boyd, Ghaoui, Feron and Balakrishnan, 1994). Therefore an affine TS model can be regarded as a mapping from the antecedent (input) space to a convex region (polytope) in the space of parameters of a quasi-linear system. This fact makes TS model attractive in stability analysis. There are a number of researches concerning the application of T-S models in real world systems. Readers please refer to (Ollero and Cuesta, 1998) for detail information. Also, the quasi-linear structure of (2.9) enable us to solve control problems by utilising linear control techniques. In this thesis we choose TS model for fuzzy modelling and control.

2.3 Constructing fuzzy models

Generally speaking, to develop a fuzzy model requires the following decisions (Babuska, 1996):

1. Choose the type of fuzzy model.
2. Choose the inference and defuzzification methods and the particular fuzzy logic or set-theoretic operators.
3. Develop the knowledge base, ie., the rules and the membership functions.
4. Apply data to derive the model and validate it.

And also, a wide range of topics such as how to use expert knowledge and how to use experiment data are involved. Here only discuss a few typical basic approaches to construct TS models directly from experimental data. The main approaches investigated are the operating point space (also referred to as *input space* hereafter in this thesis) partitioning and the consequent local model identification.

Roughly the input space partitioning approaches can be divide into two categories: data clustering, and lattice partitioning of the input space.

2.3.1 Clustering

The objective of clustering is to partition the data set into c clusters. For the time being, assume that c is known, based on prior knowledge of the plant, for instance. There are two characteristic clustering methods in fuzzy modelling, ie., *hard clustering* (Lewis, 1990; Jang, Sun and Mizutani, 1997) and *fuzzy clustering* (Jang, Sun and Mizutani, 1997; Kim, Park, Ji and Park, 1997). In the following we briefly describe these two techniques.

2.3.1.1 Hard clustering (HCM algorithm)

The *hard c-means* (HCM) algorithm tries to locate clusters in the multi-dimensional input space. The goal is to assign to each point in the input space to a particular cluster. The basic approach is as follows (Lewis, 1990).

1. Manually seed the algorithm with c clusters, one for each cluster we are seeking. This requires prior information from the outside world of the number of different clusters into which the points are to be divided; thus the algorithm belongs to the class of *supervised* algorithms.

2. Each point is assigned to the cluster centre nearest to it.

3. A new cluster centre is computed for each class by taking the mean values of the coordinates of the point assigned to it.

4. If not complete according to some stopping criterion, return to step 2.

Some additional rules can be added to remove the necessity of knowing precisely how many clusters there are. The rules allow nearby clusters to merge and clusters which have large standard deviations in coordinates to split.

The hard c-means algorithm is based on a c-partition of the data space Z into a family of clusters $\{C_i\}, i = 1, 2, ..., c$, where the following set-theoretic equations apply,

$$\begin{aligned} \bigcup_{i=1}^{c} C_i &= Z, \\ C_i \bigcap C_j &= \emptyset, \ \forall i \neq j, \\ \emptyset \subset \ C_i \ &\subset Z, \ \forall i, \end{aligned}$$

where $\emptyset$ denotes the empty set. The set $Z = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$ is a finite set with N points, and c is the number of clusters such that $2 \leq c \leq N$.

Formally, the c-means algorithm finds a centre in each cluster, minimizing an objective function based on a distance measure. The objective function depends on the distances between vectors $\mathbf{z}_k$ and cluster centre $\mathbf{c}_i$, and when the *Euclidean distance* is chosen as the distance function, the expression for the objective function is,

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} (\sum_{k,\mathbf{z}_k \in C_i} \|\mathbf{z}_k - \mathbf{c}_i\|^2), \tag{2.10}$$

where J_i is the objective function within cluster C_i.

The partitioned clusters are typically defined by a $c \times N$ binary characteristic matrix M, called the *membership matrix*, where each element m_{ik} is 1 if the kth data point $\mathbf{z}_k$ belongs to cluster C_i, and 0 otherwise. Since a data point can only belong to one cluster, the membership matrix M has the properties:

- the sum of each column is one, and
- the sum of all elements is N.

If the cluster centers $\mathbf{c}_i$ are fixed, the m_{ik} that minimize J_i can be derived as

$$m_{ik} = \begin{cases} 1 & \text{if } \|\mathbf{z}_k - \mathbf{c}_i\|^2 \leq \|\mathbf{z}_k - \mathbf{c}_j\|^2, \ \forall j \neq i \\ 0 & \text{otherwise.} \end{cases}$$

That is, $\mathbf{z}_k$ belongs to cluster C_i if $\mathbf{c}_i$ is the closest centre among all centres. If alternatively m_{ik} is fixed, then the optimal centre $\mathbf{c}_i$ that minimizes (2.10) is the mean of all vectors in cluster C_i,

$$\mathbf{c}_i = \frac{1}{|C_i|} \sum_{k,\mathbf{z}_k \in C_i} \mathbf{z}_k,$$

where $|C_i|$ is the number of objects in C_i, and the summation is an element-by-element summation of vectors.

The algorithm is iterative, and there is no guarantee that it will converge to an optimum solution. The performance depends on the initial positions of the cluster centres, and it is advisable to employ some method to find good initial cluster centres. A major drawback of HCM is the way in which it treats the boundary points. Because boundary data points may represent patterns with a mixture of properties of data in more than one clusters, they cannot be fully assigned to either of these clusters, or do they constitute a separate class. This shortcoming can be alleviated by using fuzzy clustering as shown in the following.

2.3.1.2 Fuzzy clustering (FCM algorithm)

It is reasonable to assume that points in the middle region between two cluster centres have a gradual membership of both clusters. The *fuzzified c-means algorithm* (Jang, Sun and Mizutani, 1997) allows each data point to belong to a cluster to a degree specified by a membership grade, and thus each point may belong to several clusters.

The *fuzzy c-means* (FCM) algorithm partitions a collection of N data points specified by m-dimensional vectors $\mathbf{z}_k$ $(k = 1,2,...,N)$ into c fuzzy clusters, and finds a cluster centre in each, minimizing an objective function. Fuzzy c-means is different from hard c-means, mainly because it employs fuzzy partitioning, where a point can belong to several clusters with degree of membership. To accommodate the fuzzy partitioning, the membership matrix M is allowed to have elements in the range [0, 1]. A point's total membership of all clusters, however, must always be equal to unity to maintain the properties of the M matrix mentioned above. The objective function is a generalisation of (2.10),

$$J(M,\mathbf{c}_1,...,\mathbf{c}_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\sum_{k=1}^{N} m_{ik}^q d_{ik}^2,$$

where m_{ik} is a membership between 0 and 1, $\mathbf{c}_i$ is the centre of fuzzy cluster C_i, $d_{ik} = \|\mathbf{z}_k - \mathbf{c}_i\|$ is the Euclidean distance between the ith cluster centre and kth data point, and $q \in [1,\infty)$ is a weighting exponent. There are two necessary conditions for J to reach a minimum,

$$\begin{aligned} \mathbf{c}_i &= \frac{\sum_{k=1}^{N} m_{ik}^q \mathbf{z}_k}{\sum_{k=1}^{N} m_{ik}^q}, \\ m_{ik} &= \frac{1}{\sum_{j=1}^{c} (\frac{d_{ik}}{d_{jk}})^{2/(q-1)}}. \end{aligned}$$

The algorithm is simply an iteration through the preceding two conditions. The cluster centres can alternatively be initialised before carrying out the iterative procedure. The algorithm may not converge to an optimum solution and the performance depends on the initial clusters centres, just as in the case of the hard c-means algorithm.

FCM is also a supervised algorithm, because it is necessary to tell it how many clusters to look for. If c is not known beforehand, it is necessary to apply an *unsupervised* algorithm such as *subtractive clustering*. Subtractive clustering is based on a measure of the density of data points in the input space (Jang, Sun and Mizutani, 1997). The idea is to find regions in the input space with high densities of data points. The point with the highest number of neighbours is selected as centre for a cluster. The

data points within a prespecified, fuzzy radius are then removed (subtracted), and the algorithm looks for a new point with the highest number of neighbours. This continues until all data points are examined. In the following we look at the algorithm in more detail.

Consider the set of Z given above. Since each data point is a candidate for a cluster centre, a density measure at data point $\mathbf{z}_k$ can be defined as

$$D_k = \sum_{j=1}^{N} \exp(-\frac{\|\mathbf{z}_k - \mathbf{z}_j\|}{(r_a/2)^2}),$$

where r_a is a positive constant. A data point will have a high density value if it has many neighbouring data points. Only the fuzzy neighbourhood within the radius r_a contributes to the density measure.

After calculating the density measure for each data point, the point with the highest density is selected as the first cluster centre. Let $\mathbf{z}_{c_1}$ be the point selected and D_{c_1} is its density measure. Next, the density measure for each data point $\mathbf{z}_k$ is revised by the formula

$$D'_k = D_k - D_{c_1} \exp(-\frac{\|\mathbf{z}_k - \mathbf{z}_{c_1}\|}{(r_b/2)^2}),$$

where r_b is a positive constant. Therefore, the data points near the first cluster centre $\mathbf{z}_{c_1}$ will have significantly reduced density measures, thereby making the points unlikely to be selected as the next cluster centre. The constant r_b defines a neighbourhood to be reduced in density measure. It is normally larger than r_a to prevent closely spaced cluster centres; typically $r_b = 1.5r_a$.

After the density measure for each point is revised, the next cluster centre $\mathbf{z}_{c_2}$ is selected and all the density measure are revised again. The process is repeated until a sufficient number of cluster centres are generated.

2.3.1.3 Identification of consequent submodels

Normally clustering methods are not concerned with the problem of identification of consequent submodels. This is one of the major weakness of clustering approaches used for model construction, since it is natural to have a dual process of data (model) structuring or construction (ie., clustering here) together with model determination, since there is clearly a trade off between the number of clusters, or models, model accuracy and model complexity.

In some cases the submodel identification problem can be incorporated in the clustering. One of the special cases is that the structure of the submodels is *a prior*

determined, and therefore the task reduces to parameter estimation. This idea can be illustrated by a special fuzzy clustering technique called *product space clustering* applied to a simple model structure.

A simple and practically useful parameterisation of (2.7) is an affine form, resulting in the following rules:

$$\textbf{IF } \phi \text{ is } A_i \textbf{ then } y_i = \mathbf{a}_i^T \phi + b_i. \quad i = 1,...,L, \tag{2.11}$$

where $\mathbf{a}_i$ is a parameter vector and b_i is a scalar offset, $\phi \in \Re^d$. This is the affine TS model defined in (2.9). We will now explain how fuzzy clustering can be used for constructing affine TS fuzzy models from data. The approach, proposed by Babuska and Verbruggen (1995a), is based on an assumption that the identification data is a representative sample of the regression surface (surface defined by $y = f(\phi)$). In practice, this may not be true as there may be data sparsity in serial regions of the data space, and some form of regularisation may be needed in modelling. A fuzzy clustering algorithm based on an adaptive distance measure is applied to partition this data into several clusters. These clusters are projected onto the antecedent variables to form the membership functions and from the fuzzy covariance matrices of the clusters the consequent parameters are derived. Cluster validity measures and cluster merging techniques can be applied to find an appropriate number of rules. In the following we briefly describe this method.

1. The Gustafson-Kessel's algorithm

Suppose a set of N data pairs $(\phi_j, y_j), j = 1,...,N$ is available. Denoting $\mathbf{z}_j = [\phi_j, y_j]^T$ we can write this data set in a matrix form

$$Z = [\mathbf{z}_1,...,\mathbf{z}_N], \quad \mathbf{z}_j \in \Re^{d+1}.$$

The vectors $\mathbf{z}_j$ will be partitioned into c clusters with prototypes $\mathbf{v}_i = [v_{i,1},...,v_{i,d+1}]^T \in \Re^{d+1}, i = 1,...,c$. The c-tuple of the cluster prototypes is denoted as $V = [\mathbf{v}_1,...,\mathbf{v}_c]$. The partitioning of the data is defined by means of a *fuzzy partition matrix*

$$U = [\mu_{ij}]_{c \times N},$$

where $\mu_{ij} \in [0,1]$ represents the membership degree of the data vector $\mathbf{z}_j$ in the ith cluster with the prototype $\mathbf{v}_i$. The Gustafson-Kessel's (GK) algorithm (Gustafson and Kessel, 1979) finds the partition matrix and the cluster prototypes by minimizing the following objective function:

$$J(Z,V,U) = \sum_{i=1}^{c} \sum_{j=1}^{N} \mu_{ij}^m d^2(\mathbf{z}_j, \mathbf{v}_i),$$

subject to

$$\begin{aligned} \textstyle\sum_{i=1}^{c} \mu_{ij} &= 1, \quad j = 1,...,N \text{ and} \\ 0 < \textstyle\sum_{j=1}^{N} \mu_{ij} &< N \quad i = 1,...,c, \end{aligned}$$

where $m > 1$ is a parameter that controls fuzziness of the clusters, with higher values of m the clusters overlap more. The shape of the clusters is determined by the particular distance measure $d(\mathbf{z}_j, \mathbf{v}_i)$ involved. Gustafson and Kessel (1979) gives an *adaptive distance measure*:

$$d^2(\mathbf{z}_j, \mathbf{v}_i) = (\mathbf{z}_j - \mathbf{v}_i)^T M_i (\mathbf{z}_j - \mathbf{v}_i), \tag{2.12}$$

where M_i is a positive definite matrix adapted according to the actual shapes of the individual clusters approximately described by the *cluster covariance matrices* F_i:

$$F_i = \frac{\sum_{j=1}^{N} \mu_{ij}^m (\mathbf{z}_j - \mathbf{v}_i)(\mathbf{z}_j - \mathbf{v}_i)^T}{\sum_{j=1}^{N} \mu_{ij}^m}. \tag{2.13}$$

M_i is calculated as a normalized inverse of the cluster covariance matrix F_i:

$$M_i = \det(F_i)^{\frac{1}{d+1}} F_i^{-1}. \tag{2.14}$$

Algorithm:

Given a data set Z, the number of clusters c and an initial (random) partition U, the cluster prototypes V and the final partition matrix U are found by repeating the following steps:

1. Compute $\mathbf{v}_i = \frac{\sum_{j=1}^{N} \mu_{ij}^m \mathbf{z}_j}{\sum_{j=1}^{N} \mu_{ij}^m}$.
2. Calculate F_i using (2.13).
3. Compute the matrices M_i using (2.14).
4. Calculate $d^2(\mathbf{z}_j, \mathbf{v}_i)$ as given in (2.12).
5. update the fuzzy partition matrix U : $\mu_{ij} = \frac{d(\mathbf{z}_j, \mathbf{v}_i)^{-2/(m-1)}}{\sum_{i=1}^{c} d(\mathbf{z}_j, \mathbf{v}_i)^{-2/(m-1)}}$
   if $d^2(\mathbf{z}_j, \mathbf{v}_i) = 0$ for some $i = k$, set $\mu_{kj} = 1$ and $\mu_{ik} = 0, \forall i \neq k$.

until a specified convergence criterion is satisfied, *e.g.* $\|U_l - U_{l-1}\| < \varepsilon$ where l is the iteration step and ε is a termination tolerance.

After convergence, the partition matrix U, the cluster prototypes $\mathbf{v}_i$ and covariance matrices $F_i, i = 1,...,c$ are obtained.

2. Extracting rules from the clusters

Suppose $\Phi_i = [\varphi_{i,1}, ..., \varphi_{i,d+1}]$ is the normal eigenvector corresponding to the smallest eigenvalue of F_i. Recalling that $\mathbf{v}_i$ is the cluster prototypical point we can directly write an implicit form of the TS consequents:

$$\Phi_i^T \cdot ([\phi, y] - \mathbf{v}_i) = 0.$$

The parameters $\mathbf{a}_i$ and b_i of the explicit form (2.11) are

$$\mathbf{a}_i = -\frac{1}{\varphi_{i,d+1}}[\varphi_{i,1}, ..., \varphi_{i,d}]^T, \; b_i = -\frac{1}{\varphi_{i,d+1}}\Phi_i^T \cdot \mathbf{v}_i.$$

and thus the clusters found by the GK algorithm can be represented by a set of TS rules (2.11).

The antecedent fuzzy sets A_i are obtained by projecting the fuzzy set defined pointwise in the product space onto the regression vector space:

$$\mu_{A_i}(\phi_l) = \max_{j:\phi_j=\phi_l, j>l} \mu_{ij}, \quad \forall l \in \{1, ..., N\}.$$

3. Determine the number of rules

The number of clusters must be specified before clustering. Empirical relations among the number of clusters c, the dimension of the regression vector d and the number of data points N have been suggested by Jain and Dubes (1988). If no particular knowledge about the type of the process nonlinearity is available, automated procedures for determining the number of rules can be applied. In connection with fuzzy clustering two main approaches are used, i.e., validity measures and compatible cluster merging:

- *Validity measures* are numerical indicators that assess the qualities of the clusters. The data must be clustered several times, each time with a different number of clusters. The number of clusters that minimize (maximize) the validity measure is selected. A dedicated validity measure for the clustering-based identification of nonlinear systems was proposed by Babuska and Verbruggen (1995b), which combines a measure of the cluster flatness with the mean prediction error. This approach conceptually resembles the use of information criteria in linear system identification (Akaike, 1974) and can also be used for selecting the structure of the model.

- *Cluster merging* approaches start with a high number of clusters and proceed by gradually merging similar clusters (Krishnapuram and Freg, 1992; Kaymak and Babuska, 1995). The initial number of clusters must be set sufficiently high such that the nonlinearity of the regression hypersurface can be captured accurately enough. The number of clusters is iteratively reduced by merging clusters that are sufficiently close and approximately parallel. Two clusters C_i and C_j are approximately parallel if the dot product of their normal vectors is close to one:

$$|\Phi_i \cdot \Phi_j| > k_1, \quad k_1 \text{ close to } 1.$$

The distance of the clusters is measured as the Euclidean distance of the cluster prototypical points:

$$\|\mathbf{v}_i - \mathbf{v}_j\| < k_2, \quad k_2 \text{ close to } 0.$$

2.3.2 Lattice partitioning

In this category the TS model in (2.7) is decomposed into the following form

$$\begin{aligned} R_i: \quad & \textbf{If } z_1 \text{ is } A_{1i} \textbf{ AND } z_2 \text{ is } A_{2i} \textbf{ AND } \ldots \textbf{ AND } z_d \text{ is } A_{di} \\ & \textbf{then } \mathbf{y}_i = \mathbf{f}_i(\mathbf{x}), \quad i = 1,2,\ldots L, \end{aligned} \tag{2.15}$$

where univariate fuzzy sets are formed in each input axis which independent of fuzzy sets defined on other input axis. Multivariate fuzzy sets are formed by taking the tensor product of the univariate fuzzy sets.

Our meaning of the term *lattice partitioning* here includes all of input axis orthogonal partitioning strategies, rather than the normal sense of lattices. Increased resolution can always be achieved by refining the input lattice by additional basis functions, again preserving the fuzzy linguistic interpretation of the resultant system. Possible methods for achieving this are shown in figure 2.2.

The K-tree and quad-tree approaches produce axis orthogonal splits, which whilst aiding the representational aspects of the fuzzy system, limits the submodel modelling capability. In the following we demonstrate the basic idea of lattice partitioning by a simple algorithm using the *Unbiasedness Criterion* (UC).

2.3.2.1 UC algorithm

The UC is provided by Sugeno and Kang (1988) for the verification of a model structure. Using the criterion, it generates an algorithm for identifying the structure and

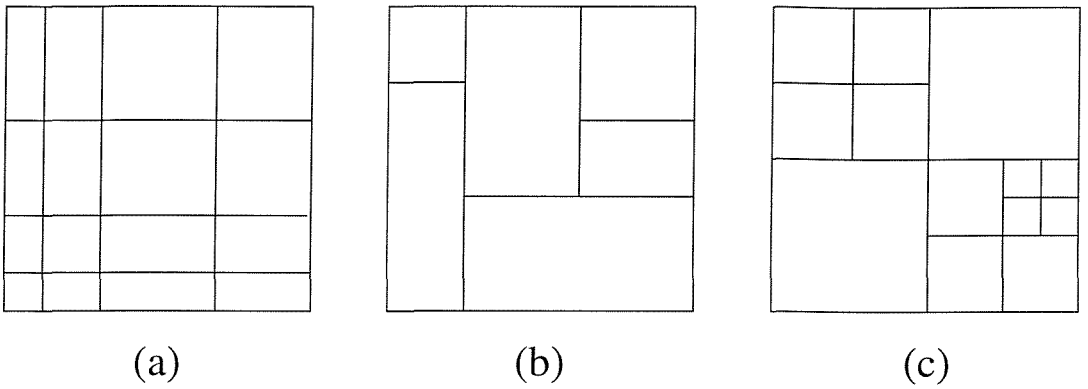

Figure 2.2: Three different lattice partitioning strategies: (a) standard lattice (b) K-tree and (c) quad tree.

parameters of a TS model from observed data. We first divide the observed data into sets N_A and N_B, and identify the consequent parameters for each set of data separately. Then the UC is calculated as

$$\mathrm{UC} = [\sum_{i=1}^{n_A} (y_i^{AB} - y_i^{AA})^2 + (y_i^{BA} - y_i^{BB})^2]^{1/2}.$$

where n_A is the number of the elements in data set N_A, y_i^{AA} the estimated output for the data set N_A from the model identified by using the data set N_A, and y_i^{AB} the estimated output for the data set N_A from the model identified by using the data set N_B.

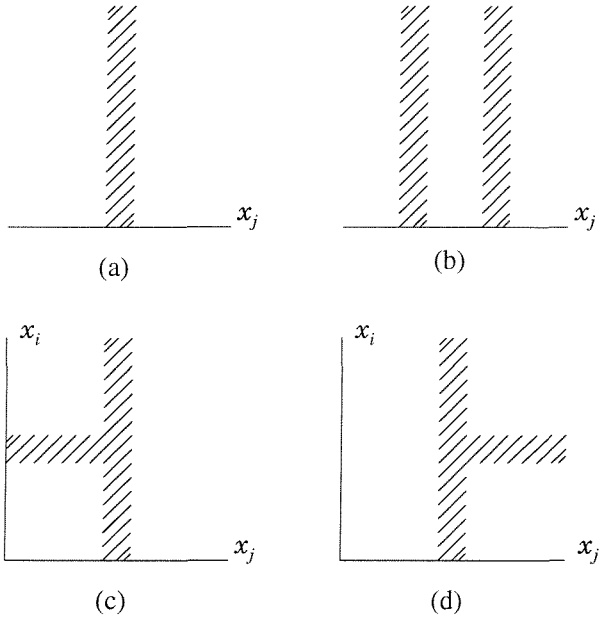

Figure 2.3: Premise structure identification

UC Algorithm

1. An ordinary linear model is identified. Its UC is calculated and written as $UC_{[1]}$.

2. A fuzzy model consisting of two fuzzy implications is constructed by first dividing the range of x_1 into two fuzzy subspaces. Then the premise parameters, the consequent structure and parameters are identified. The UC of the model is calculated. Similarly, a fuzzy model dividing the range of x_i for $i = 2,...,k$ is identified and its UC is calculated. Among the k models, one with the least UC is picked up. Its premise structure and UC are written as $ST_{[2]}$ and $UC_{[2]}$, respectively.

3. Suppose that the input variable x_j is found to be put into the premises at the stage 2. Then $ST_{[2]}$ is as shown in Figure 2.3(a). The three possible constructions of premise structure are shown in figure 2.3(b)—(d).

4. the stages after the stage 3 are similar to the stage 3. Whether the process is terminated or not is decided by comparing $UC_{[i]}$ with $UC_{[i-1]}$

Figure 2.4 illustrates the algorithm for identifying the structure and parameters of a TS model. From figure 2.3 we can see that the algorithm is one kind of K-tree methods.

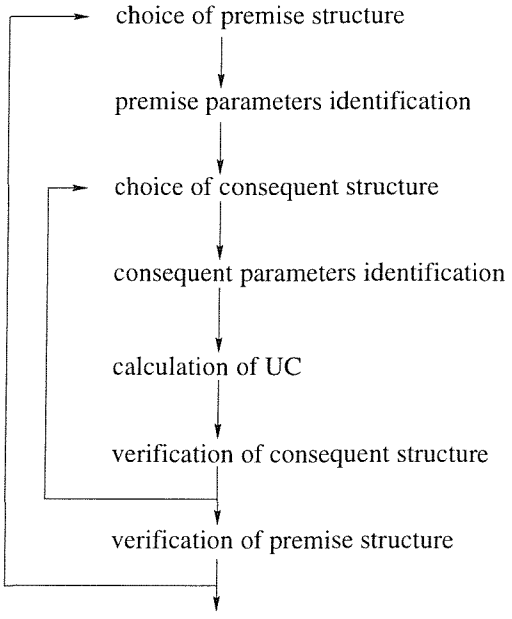

Figure 2.4: The UC Algorithm

Other similar algorithms based on lattice partitioning include the so called *local Linear Model Trees* method described in Nelles, Sinsel and Isermann (1996) which is for identification of models with local basis function networks structure, and the *semi-empirical decomposition of operating point space* proposed by Johansen and Foss (1995) which is developed within a more general local modelling framework.

2.3.2.2 Local model identification

Johansen and Foss (1995) gives a typical general technique which integrates the local model identification problem with the partitioning of the operating point space. Suppose we have a decomposition of operating space into L disjoint regions:

$$Z = \bigcup_{i \in I_L} Z_i.$$

for some index set $I_L = \{i_1, ..., i_L\}$. A model structure of (2.8) based on above decomposition is defined as

$$S_L = \{(Z_i, \bar{w}_i, \mathbf{f}_i(\mathbf{x}, \theta_i))\}_{i \in I_L}. \tag{2.16}$$

where θ_i is the parameter of the ith local model. Assume that the data observations are bounded. Then the system's operating range Z can be approximated by the d-dimensional box

$$Z_1 = [z_{1,1}^{\min}, z_{1,1}^{\max}] \times \cdots \times [z_{1,d}^{\min}, z_{1,d}^{\max}].$$

Next we consider the problem of decomposing Z_1 into regions.

Consider the possible decompositions $Z_1 = Z_{11} \cup Z_{12}$. We restrict these possibilities by the constraint that the splitting boundary is a hyper-plane orthogonal to one of the natural basis vectors of $\Re^d$, i.e.

$$Z_{11} = \{\mathbf{z} \in Z_1 | z_{d_1} < \zeta_1\}$$
$$Z_{12} = \{\mathbf{z} \in Z_1 | z_{d_1} \geq \zeta_1\}$$

for some dimension index $d_1 \in \{1, ..., d\}$ and splitting point $\zeta_1 \in [z_{1,d_1}^{\min}, z_{1,d_1}^{\max}]$. Local model validity functions for the two regions are defined by the recursion

$$\bar{w}_{11}(\mathbf{z}) = \bar{w}_1(\mathbf{z}) b(z_{d_1} - \bar{z}_{11,d_1}; \lambda_{11})$$
$$\bar{w}_{12}(\mathbf{z}) = \bar{w}_1(\mathbf{z}) b(z_{d_1} - \bar{z}_{12,d_1}; \lambda_{12})$$

where $\bar{z}_{i,d_1} = 0.5(z_{i,d_1}^{\min} + z_{i,d_1}^{\max})$ for $i \in \{11, 12\}$ is the center point of Z_i in the d_1-direction. The function $b(r; \lambda)$ is a scalar basis function with scaling parameter λ,

and the *local model validity function* (degree of fulfillment) associated with the region Z_1 is $\bar{w}_1(\mathbf{z}) = 1$. The scaling parameters are chosen by considering the overlap between the local model validity functions. For $i \in \{11, 12\}$, we choose $\lambda_i = 0.5\gamma(z_{i,d_1}^{\max} - z_{i,d_1}^{\min})$ where γ is a design parameter that typically takes a value between 0.25 and 2.0. There will be almost no overlap when $\gamma = 0.25$, and large overlap when $\gamma = 2.0$. For each dimension index $d_1 \in \{1,...,d\}$ we represent the interval $[z_{1,d_1}^{\min}, z_{1,d_1}^{\max}]$ by a finite number of L_1 points uniformly covering the interval. Now $d_1, \zeta_1, \mathbf{f}_1$, and $\mathbf{f}_2$ defines a new model structure, where the region Z_1 is decomposed according to the dimension index d_1 at the point ζ_1, and the two local model structures are $\mathbf{f}_1$ and $\mathbf{f}_2$. Formally, the set of candidate model structures $\mathcal{S}_n$ with n ($n = 1, 2, ...$) regions is given by

$$
\begin{aligned}
\mathcal{S}_1 &= \{\{(Z_1, \bar{w}_1, \mathbf{f}_j)\}; j \in \{1,2,...,N_L\}\} \\
\mathcal{S}_2 &= \{\{(Z_{11}^i, \bar{w}_{11}^i, \mathbf{f}_j), (Z_{12}^i, \bar{w}_{12}^i, \mathbf{f}_k)\}; i \in \{1,2,...,d \cdot N_1\}, j,k \in \{1,2,...,N_L\}\} \\
\mathcal{S}_3 &= \{\{(Z_{11}^i, \bar{w}_{11}^i, \mathbf{f}_j), (Z_{121}^m, \bar{w}_{121}^m, \mathbf{f}_k), (Z_{122}^m, \bar{w}_{122}^m, \mathbf{f}_n)\}; \\
&\qquad i,m \in \{1,2,...,d \cdot N_1\}, j,k,n \in \{1,2,...,N_L\}\} \\
&\quad \cup \{\{(Z_{111}^m, \bar{w}_{111}^m, \mathbf{f}_k), (Z_{112}^m, \bar{w}_{112}^m, \mathbf{f}_m), (Z_{12}^i, \bar{w}_{12}^i, \mathbf{f}_j)\}; \\
&\qquad i,m \in \{1,2,...,d \cdot N_1\}, j,k,n \in \{1,2,...,N_L\}\} \\
\mathcal{S}_4 &= ...
\end{aligned}
$$

The model structure set is now

$$\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 \cup ...$$

which is illustrated as a search tree in figure 2.5 (from Johansen and Foss (1995)). Now the structure identification problem can be looked upon as a multi-step decomposition process, where at each step one region from the previous step is decomposed into two sub-regions. Such an approach will lead to a sequence of model structures $S_1, S_2, ..., S_n$ where the model structure S_{i+1} has more degree of freedom than S_i.

1. Model structure identification criteria

Let a model structure S of the form (2.16) be given. Together with the admissible parameter set θ_S, S induces a model set

$$\mathcal{M}_S = \{M = (S, \theta); \theta \in \theta_S\}.$$

Introduce the notation

$$
\begin{aligned}
y(t) &= y^*(\mathcal{D}_{t-1}) + e(t), \\
\varepsilon(t|S,\theta) &= y(t) - \hat{y}(t|\mathcal{D}_{t-1}, S, \theta),
\end{aligned}
$$

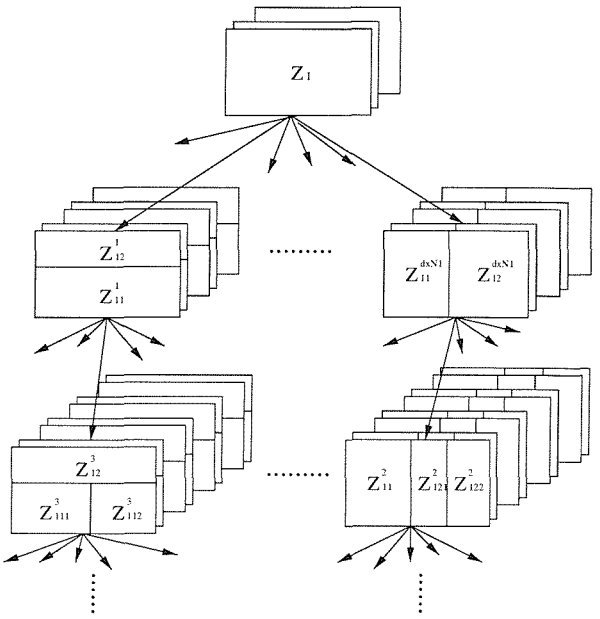

Figure 2.5: Model structure search tree illustrating possible decompositions into regions and choice of local model structures. Each level in the tree corresponds to the possible decompositions into one more region than at the previous level, i.e., the model structure sets $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ etc. The subset of model structures at each "super-node" in the tree corresponds to a fixed decomposition into regions, but different combinations of local model structures.

where $\mathcal{D}_{t-1}$ is the observation data with $t-1$ points and $y^*(\mathcal{D}_{t-1})$ is the deterministic (predictable) component of the system output, $e(t)$ is the stochastic (unpredictable) component and $\varepsilon(t|S,\theta)$ is the residual. Let $\hat{\theta}_S$ be a parameter estimate that minimize the prediction error criterion (Ljung, 1987)

$$J_S(\theta) = \frac{1}{N}\sum_{t=1}^{N} \text{trace}(\varepsilon(t|S,\theta)\varepsilon^T(t|S,\theta)). \tag{2.17}$$

Let an unknown *future* data sequence be denoted by $\mathcal{D}_t^*$, and assume $\mathcal{D}_t^*$ and $\mathcal{D}_N$ are uncorrelated. Moreover, let $E_{\mathcal{D}}$ and $E_{\mathcal{D}^*}$ denote expectations with respect to $\mathcal{D}_N$ and $\mathcal{D}_t^*$, respectively. The prediction error is given by (Ljung, 1987)

$$\varepsilon^*(t|S,\hat{\theta}_S) = y^*(\mathcal{D}_{t-1}^*) - \hat{y}(t|\mathcal{D}_{t-1}^*, S, \hat{\theta}_S(\mathcal{D}_N)) + e(t).$$

The expected squared prediction error is defined by

$$\Sigma(S) = E_{\mathcal{D}^*}E_{\mathcal{D}}\varepsilon^*(t|S,\hat{\theta}_S(\mathcal{D}_N))\varepsilon^*(t|S,\hat{\theta}_S(\mathcal{D}_N))^T.$$

The *mean square error* (MSE) criterion is defined by

$$J_{MSE}(S) = \text{trace}(\Sigma(S)).$$

Because the probability distribution for the prediction error is unknown, J_{MSE} cannot be computed. An alternative would be to minimize the average squared *prediction error* (PE) criterion (Ljung, 1987)

$$J_{PE}(S) = \text{trace}(\frac{1}{N}\sum_{t=1}^{N}\varepsilon(t|S,\hat{\theta}_S)\varepsilon(t|S,\hat{\theta}_S)^T).$$

For finite N, J_{PE} may be a strongly biased estimate of J_{MSE}. In the following several criteria that are far better estimates of J_{MSE} than J_{PE} will be present.

1. *Final prediction error* criterion (FPE) (Akaike, 1969):

$$J_{FPE}(S) = \frac{1+p(S)/N}{1-p(s)/N}J_{PE}(S),$$

   where $p(S)$ is the effective number of parameters (degrees of freedom) in the model structure.

2. *Generalized Cross Validation* (GCV) (Craven and Wahba, 1979):

$$J_{GCV}(S) = \frac{1}{(1-p(s)/N)^2}J_{PE}(S),$$

   which is easily seen to be asymptotically equivalent to FPE, and assumes linear parameterization of the predictor.

Any one of these criteria can be applied with the structure identification algorithm presented in the following subsection.

2. Basic search algorithm

The problem is now to search the set $\mathcal{S}$ for the best possible model structure. The estimate of the parameters in model structure S is defined by

$$\hat{\theta} = arg\ \min_{\theta} J_S(\theta)$$

where $J_S(\theta)$ is defined by (2.17), and it has been assumed that a unique minimum exists. Now, the chosen structure identification criterion is written as $J'(S)$. We define for a given n

$$S_n = arg\ \min_{S\in\mathcal{S}_n} J'(S)$$

where it again has been assumed that a unique minimum exists. Consider the following extended horizon search algorithm, where the integer $n^* \geq 1$ is called the search horizon:

Search Algorithm

1. Start with the region Z_1. Let n=1.

2. At each step $n \geq 1$, find a sequence of decompositions and local model structures $S_n, S_{n+1}, ..., S_{n+n^*}$ that solves the optimization problem

$$\min_{S \in \mathcal{S}_{n+n^*}} J'(S).$$

3. Restrict the search tree by keeping the decomposition that leads to S_{n+1} fixed for the future.

4. If

$$J'(S_n) > \min_{k \in \{1,2,...,n^*\}} J'(S_{n+k})$$

then increment n and go to step 2. Otherwise, the model structure S_n is chosen.

Referring to figure 2.5, this algorithm will search the tree starting at the top, and selecting a decomposition at each level through a sequence of "locally exhaustive" searches of depth n^*. In other words, this algorithm will make an n^*-step-ahead optimal decomposition at each step, in the sense that the decomposition is optimal if there is going to be exactly n^* more decompositions. If $n^* = 1$, this is a local search algorithm.

3. Heuristic search algorithm

To reduce the number of candidates, Johansen and Foss (1995) suggests to apply the following heuristic search algorithm in the "locally exhaustive" search at the second step in the previous search algorithm:

1. At each level in the search tree, proceed with only the most promising candidates.

2. Discard the candidate decompositions that give an increase in the criterion from one level in the search tree to the next.

3. Discard the candidate decompositions that lead to regions with less data points relevant to the region than the number of degrees of freedom in the corresponding local model structure and local model validity function.

4. Use a (backward or forward) stepwise regression procedure to handle local model structure sets $\mathcal{L}$ of combinatorial nature.

2.3.3 Comments

In this section, two typical operating point space partitioning methods were presented, ie., clustering and lattice partitioning. The lattice partitioning methods are able to determine the model structure and construct models from the experimental data. But it is easy to see that within the lattice partitioning framework, when the dimension of the operating point variable is large, the number of the possible structures at each stage becomes combinatorially large. That is to say, these kind of algorithms share the common problem of *the curse of dimensionality*. Clustering, on the other hand, can avoid the curse of dimensionality. But for a general local model structure present clustering methods are unable to include the problem of identification of consequent submodels. Also, most clustering algorithms need *a priori* known number of clusters.

Besides the above mentioned techniques, another popular modelling algorithm in this field is *Adaptive Spline Modelling of Observation Data* (ASMOD) (Kavli, 1993; Kavli, 1994; Brown and Harris, 1994). This is covered in other theses by ISIS researches, see eg., Bossley (1997). ASMOD is an important attempt to solve the problem by reproducing the internal structure and dependencies contained in the training data whilst retaining transparency. But It still has not thoroughly overcome the curse due to its inherent lattice partitioning nature.

Nevertheless, using the techniques presented in this section or the new techniques in next chapter, we are able to construct TS models for the systems (2.1) - (2.6). Next section lists a number of typical TS models which can be derived by these techniques from experiment data of systems (2.1) - (2.6) and will be used in subsequent chapters.

2.4 Some types of TS models

There are many alternatives of the local models in TS model (2.7) or (2.8). We can choose either continuous time models or discrete time models, and also either input-output models or state space models. In the following we list some types of TS models corresponding to some of the systems (2.1)—(2.6) which can be established by the techniques described in section 2.3.

Discrete time input-output TS model

$$\mathbf{y}_{t+1} = \sum_i w_i(\mathbf{x}_t)\mathbf{f}_i(\mathbf{x}_t). \tag{2.18}$$

Discrete time state space TS model

$$\mathbf{x}(t+1) = \sum_i w_i(\mathbf{x}(t))\mathbf{f}_i(\mathbf{x}(t)). \tag{2.19}$$

Continuous time state space TS model

$$\dot{\mathbf{x}}(t) = \sum_i w_i(\mathbf{x}(t))\mathbf{f}_i(\mathbf{x}(t)). \tag{2.20}$$

Affine continuous time state space dynamic TS model

$$\dot{\mathbf{x}}(t) = \sum_i w_i(\mathbf{x}(t))\mathbf{f}_i(\mathbf{x}(t)) + \sum_i w_i(\mathbf{x}(t))\mathbf{g}_i(\mathbf{x}(t))\mathbf{u}(t). \tag{2.21}$$

Chapter 3 will use the discrete time TS models (2.18) and (2.19) to demonstrate the new modelling techniques. Chapter 5 will use model (2.20) for local stability analysis. The affine continuous time state space dynamic TS model (2.21) is affine in control signal which will be used in chapter 6 for controller design.

2.5 Quasi-linear system models with neural network parameters

Consider a general nonlinear dynamical system given by (2.1). In the design of nonlinear control systems linearisation about a fixed, known operating point is a standard solution as it leads to a *locally* linear model for which a controller can be synthesized by classical linear control design methods. For many practical processes (e.g., aircraft gas turbines or ship dynamics), the operating point varies either as a function of independent parameters (such as mach number and altitude for an aero-gas turbine) or more usually, as a function of the system states. In the light of this observation, (2.1) can be represented by quasi-linear system model of the form (Billings and Voon, 1987; Chen and Billings, 1989; Johansen, 1994a; Johansen and Foss, 1993; Priestley, 1988; Sastry and Isidori, 1989)

$$\begin{aligned} \mathbf{y}(t+1) &= a_1(O_t)\mathbf{y}(t) + \cdots + a_n(O_t)\mathbf{y}(t-n+1) \\ &\quad + b_1(O_t)\mathbf{u}(t-d) + \cdots + b_m(O_t)\mathbf{u}(t-d-m+1) \end{aligned} \tag{2.22}$$

where $a_i(O_t)$ and $b_j(O_t)$ $(i = 1,2,\cdots,n;\ j = 1,2,\cdots,m)$ are *a priori* unknown functions of the measured operating point O_t. The existence of representation (2.22) depends on $\mathbf{g}(\cdot)$ being first order differentiable with respect to its arguments $\mathbf{x}_t$ defined in section 2.1.1 (Billings and Voon, 1987; Chen and Billings, 1989; Sastry and Isidori,

1989). This is an assertion since $\mathbf{g}(\cdot)$ is *a priori* unknown in adaptive control, but for most practical processes it applies. When (2.22) is a single-input single-output (SISO) system, various special cases have been studied including:

(i) The nonlinear function $g(\cdot)$ can be decomposed (Wang, Liu, Harris and Brown, 1995) such that

$$g(\cdot) = \sum_{i=0}^{n-1} f_i(y(t-i)) + \sum_{j=0}^{m-1} g_j(u(t-d-j))$$

for all $f_i(\cdot)$, $g_j(\cdot)$ first order differentiable.

(ii) The operating point O_t is independent of current value of $\mathbf{x}_t$, and is a function of independent measurable variables (Wang, Liu and You, 1991; Wang, Wang, Brown and Harris, 1996). Under assumptions concerning smoothness properties of the *unknown* parameters a_i, b_j, Taylor series expansions have been utilised to identify the unknown system, which is in turn controlled by an adaptive controller under the *certainty equivalence principle* (Xie and Evans, 1984). The high order smoothness properties of the Taylor series approach greatly limit this method.

(iii) Other special cases that have been well researched via conventional statistical based methods (Priestley, 1988) include:

(a) $\{a_i(\cdot)\}$ are constants and

$$b_j(\cdot) = c_j + \sum_{i=1}^{p} b_{ij} y(t-i); \quad j = 1, \cdots, m$$

for c_j, b_{ij} constants. Then the system (2.22) is bilinear.

(b) $\{a_i(\cdot)\}$, $\{b_j(\cdot)\}$ are constants, then (2.22) is an autoregressive moving average (ARMA) process.

(c) $\{a_i(\cdot)\}$, $\{b_j(\cdot)\}$ depend only on t, then (2.22) is linear and non-stationary.

2.5.1 Model structure

For the general class of operating point dependent quasi-linear model (2.22), if O_t is independent of the system input and output, the multiple differentiability, local conditioning, and input boundedness conditions can be relaxed, even when the parameters $\{a_i(\cdot)\}$, $\{b_j(\cdot)\}$ are *a priori* unknown. It is well known (Brown and Harris, 1994;

Harris, Moore and Brown, 1993) that unknown continuous nonlinear function (such as $a_i(\cdot)$, $b_j(\cdot)$ and $\mathbf{f}(\cdot)$) defined on a bounded closed set can be approximated with arbitrary accuracy by various neural networks, avoiding the restrictions and limitations of the Taylor series approach.

If (2.22) is SISO, the nonlinear coefficients can be expressed via *associative memory networks* (Wang, Brown and Harris, 1996) as (from now on we assume $d = 0$ for simplicity)

$$N_{a_i}(O_t) = \sum_{k=1}^{l} w_{ik}^a \phi_k(O_t) \quad i = 1, \cdots, n \tag{2.23}$$

$$N_{b_j}(O_t) = \sum_{k=1}^{l} w_{jk}^b \phi_k(O_t) \quad j = 1, \cdots, m \tag{2.24}$$

where $\{w_{ik}^a\}$ and $\{w_{jk}^b\}$ are the unknown weights of the neural networks, and $\{\phi_k(\cdot)\}$ are known basis functions (including B-spline, CMAC, and Radial basis functions with prespecified centres (Brown and Harris, 1994)) and l is a known integer ($l \gg n, m$). The motivation for using this type of architecture is the assertion that the space of measurable operating points is generally considerably smaller than a conventional nonlinear autoregressive moving average (NARMAX) system input space; in practice this is generally true. If the nonlinear coefficient representations (2.23) and (2.24) are based on B-spline polynomials (Brown and Harris, 1994), then the resulting local models (see equations (2.25) and (2.26) below) are a generalised neurofuzzy modeling extension (Harris, Wu and Feng, 1997) of the Takagi-Sugeno fuzzy input/output model (Jain, 1997). These equivalence conditions are quite mild, in that the fuzzy operations are restricted to the algebraic sum/product operations, defuzzification is by centre of area, and the membership functions are B-splines with a partition of unity ($\sum_k \phi_k(x) = 1$) to provide normalisation. The number of parameters or weights $\{w_{ik}^a\}$ and $\{w_{jk}^b\}$ used in the representations (2.23) and (2.24) can be pruned by automatic construction algorithms such as ASMOD, that have been derived for solving the curse of dimensionality in neurofuzzy systems (Harris, Wu and Feng, 1997). In practice this reduces the numbers of parameters from exponential in the input dimension $(n+m)$ to almost linear in $(n+m)$. The system (2.22) can be rewritten using (2.23) and (2.24) as

$$y(t+1) = \sum_{k=1}^{n} N_{a_k}(O_t) y(t-k+1) + \sum_{k=1}^{m} N_{b_k}(O_t) u(t-k+1) + \triangle f(O_t), \tag{2.25}$$

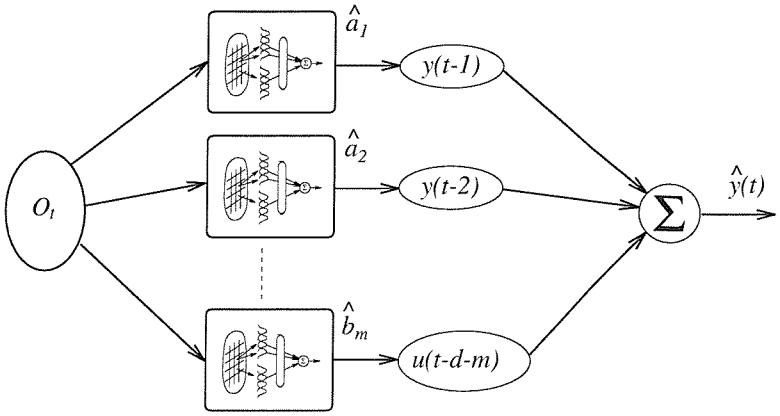

Figure 2.6: The neurofuzzy modelling architecture for the quasi-linear system (2.25).

where $\triangle f(O_t)$ is a model mismatching error:

$$\triangle f(O_t) = \sum_{k=1}^{n} (N_{a_k}(O_t) - a_k(O_t))y(t-k+1) + \sum_{k=1}^{m} (N_{b_k}(O_t) - b_k(O_t))u(t-k+1). \tag{2.26}$$

Figure 2.6 illustrates the neurofuzzy modelling architecture for the quasi-linear system (2.25).

2.5.2 Parameter estimation

If the networks can exactly model the nonlinear functions, the system is referred to as a *matching system*, whereas the system which can only approximate the nonlinear functions to a known accuracy is called a *mismatching system*. For a matching system it has been shown (Wang, Brown and Harris, 1996) that the representation (2.25) can be transformed into the form

$$y(t+1) = \Phi^T(t)\theta, \tag{2.27}$$

where

$$\theta = [w_{11}^a \cdots w_{1l}^a \cdots w_{nl}^a \, w_{11}^b \cdots w_{1l}^b \cdots w_{ml}^b]^T, \tag{2.28}$$

$$\Phi(t) = [\phi_1 y(t) \cdots \phi_l y(t) \cdots \phi_l y(t-n+1) \, \phi_1 u(t) \cdots \phi_l u(t-m+1)]^T. \tag{2.29}$$

Equation (2.27) is in the standard form of linear regression for parametric identification. Let $\hat{\theta}$ be the estimate of θ then the normalised *least mean square* (LMS)

algorithm (Brown and Harris, 1994) can be utilised:

$$\begin{aligned} \triangle\hat{\theta}(t) &= \hat{\theta}(t)-\hat{\theta}(t-1)=\eta\frac{\Phi(t-1)\varepsilon(t)}{\|\Phi(t-1)\|^2+c}, && (2.30)\\ \varepsilon(t) &= y(t)-\hat{y}(t-1), && (2.31)\\ \hat{y}(t-1) &= \Phi^T(t-1)\hat{\theta}(t-1), && (2.32) \end{aligned}$$

where $\eta \in (0,2)$ is the learning rate, c a positive arbitrarily small constant, $\hat{\theta}(0)$ initial condition. The quasi-linear model identification of the system (2.25) for model matching is summarized by:

Theorem 2.1 *(Polycarpou and Ioannou, 1992) When the normalized LMS algorithm (2.30)—(2.32) is applied to the data set* $\{y(t),u(t)\}$ *generated by original process (2.22) (single-input single-output), the equalities*

$$\begin{aligned} &\lim_{t\to\infty} \quad \frac{\varepsilon^2(t)}{\|\Phi(t-1)\|^2+c}=0,\\ &\lim_{t\to\infty} \quad \|\hat{\theta}(t)-\hat{\theta}(t-t_0)\|=0 \end{aligned}$$

hold, where t_0 *is a positive integer. Furthermore, if* $\|\Phi(t-1)\|<\infty$, *then* $\lim_{t\to\infty}(y(t)-\hat{y}(t-1))=0$.

Later in chapter 7, we will deal with the modelling and control problems of MIMO mismatching systems of general form (2.22) with known orders, in which $a_i(\cdot)$ and $b_j(\cdot)$ are unknown functions of $\mathbf{x}_t$.

2.6 Conclusions

In this chapter, a general view of existing local modelling techniques is given. The concepts and the typical methods of fuzzy modelling and a class of quasi linear operating point dependent modelling are presented. In summary, to overcome the shortcomings of clustering and lattice partitioning, it is necessary to develop new local modelling methods which are able to get rid of the curse of dimensionality, do not need to prespecify the number of local areas, and identify the local models with the process of partitioning. In next chapter, we will use the background presented in this chapter to develop such kind of methods. In chapter 7, the modelling problem of general quasi linear operating point dependent model structure (2.22) is solved together with the controller design and stability analysis.

Chapter 3

Two New Local Modelling Schemes

The purpose of this chapter is to present two new partitioning techniques for data based modelling of *a priori* unknown nonlinear dynamical system. As we have seen in chapter 2, there has been a considerable number of studies on fuzzy based modelling (Brown and Harris, 1994; Bossley, 1997). Many of them are based on lattice based partitioning of the input space and as such suffer from *the curse of dimensionality* (COD), in that as the input dimension increases the parametrisation, computation cost, training data requirements, etc. increase exponentially. The Adaptive Spline Modelling of Observational Data (ASMOD) algorithm (Kavli, 1993) was introduced to overcome this problem by decomposing a multivariable input-output mapping into a sum of lower dimensional submodels. But the number of submodels still grows very quickly when the dimension increases because of the input-space inherent lattice partitioning nature. Another main disadvantage of lattice partitioning is the problem of model discontinuities resulting in large approximation error possibly leading to instability of consequent controllers. Another typical partitioning method is clustering including *hard clustering* (Lewis, 1990; Jang, Sun and Mizutani, 1997) and *fuzzy clustering* (Jang, Sun and Mizutani, 1997; Kim, Park, Ji and Park, 1997) which have been described in chapter 2. One of the drawbacks of clustering methods is that normally the number of clusters must be specified before clustering. Also, a high density of data may not be significant because it may simply be where high numbers of samples have been taken. Perhaps more significant is the variance of the data in clusters, since high data variance suggests rapid changes in process variables.

Nonlinear systems can also be approximated by *piecewise locally linear models* via local linearisation (Smith and Johnansen, 1997; Billings and Voon, 1987). Not only can a multiple local modelling approach be more efficient in capturing the real system dynamics than a single global nonlinear model (Smith and Johnansen, 1997; Hsu,

1987), but also piecewise locally linear models are advantageous in that well-known linear control and filtering methods are directly applicable to the identified model. In a piecewise linear model, the input space is partitioned into a set of local (usually orthogonal) regions. The local models that operate on these regions are identified separately and the system output is based on a composition of the local models to obtain a good global approximation to the real system.

The approximation performance of a piecewise locally model is strongly influenced by how the input space is partitioned. To overcome the problems raised by lattice partitioning and clustering, in this chapter, firstly we provide a new fuzzy modelling technique by modifying the lattice partitioning method proposed by Tan and Yu (1996) and a method of adding new multi-dimensional membership functions to the system to form a new model (Feng and Harris, 1999a). This method is able to derive fuzzy models from data automatically, avoiding the COD problem, which is illustrated by an example.

Secondly, we introduce a new algorithm for the construction of a Delaunay input space partitioned optimal piecewise locally linear models to overcome the COD as well as generate locally linear models directly amenable to linear control and estimation algorithms (Harris, Hong and Feng, 1999). The training of the model is configured as a new mixture of experts network with a new fast decision rule derived using convex set theory. A very fast simulated reannealing (VFSR) algorithm is utilised to search for a global optimal solution of the Delaunay input space partitioning. A benchmark non-linear time series is used to demonstrate the modelling approach.

This chapter is organized as follows: Section 3.1 to section 3.4 are devoted to develop the first modelling scheme; Section 3.5 to section 3.8 are for developing the second modelling scheme. For the first method, section 3.1 presents the model structure. Section 3.2 derives the modelling algorithm. Section 3.3 proves the convergence results. Section 3.4 gives a simulation example. For the second method, section 3.5 gives an overview of the algorithm including its background. Section 3.6 presents the training algorithm of Delaunay input space partitioned piecewise linear models with the introduction of the new decision rule derived from convex set theory. Section 3.7 describes the application of VFSR algorithm to achieve a global smooth piecewise linear model via the optimisation of input space partition. Numerical examples are presented in Section 3.8. The conclusions of the whole chapter are given in Section 3.10.

3.1 A new partitioning approach for fuzzy model construction

In this section, we assume that an input sequence $\{\mathbf{u}_t\}$ is applied to an unknown nonlinear discrete-time multivariable dynamical system (2.2) generating an output sequence $\{\mathbf{y}_t\}$. Recall that in section 2.1.1 we obtained a discrete time nonlinear model of the form

$$\mathbf{y}_{t+1} = \mathbf{f}(\mathbf{y}_t, \mathbf{u}_t), \tag{3.1}$$

where $\mathbf{u}_t \in \Re^{\xi}$ and $\mathbf{y}_t \in \Re^{\eta}$ at discrete-time instant t; $\mathbf{f}(\cdot)$ is some unknown nonlinear vector function. With the definition $\mathbf{x}_t = [\mathbf{y}_t^T \ \mathbf{u}_t^T]^T \in \Re^{\gamma}, \gamma = \eta + \xi = np + mq$, (3.1) becomes (cf. (2.2) and (2.3))

$$\mathbf{y}_{t+1} = \mathbf{f}(\mathbf{x}_t). \tag{3.2}$$

3.1.1 Fuzzy model

Let a subset D in $\Re^{\gamma}$ be given. With the help of membership functions, we can readily define fuzzy quantities on D. To this end, let us suppose by some means we have a set of appropriately distributed points $\mathbf{p}_i (i = 1, 2, ...)$ in D. Then the Gaussian membership function centered at $\mathbf{p}_i$ can be defined by:

$$\mu_i(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{p}_i\|^2/\sigma_i^2}, \tag{3.3}$$

where $\mathbf{x} \in D$, $\|\cdot\|$ denotes the γ-vector norm, and σ_i is a positive scalar which determines the shape of $\mu_i(\mathbf{x})$. In the following we will use the *normalized membership functions* of the form

$$w_i(\mathbf{x}) = \frac{\mu_i(\mathbf{x})}{\sum_j \mu_j(\mathbf{x})}. \tag{3.4}$$

We regard the collection of all such membership functions $\{w_i(\mathbf{x}) | i = 1, 2, ...\}$ as defining a fuzzy quantization on D, and each $w_i(\mathbf{x})$ as a fuzzy quantity. We shall use w to identify the fuzzy quantity with the membership function $w(\mathbf{x})$, and $D_F = \{w_i | i = 1, 2, ...\}$ to denote a fuzzy quantization on D.

We use notation $\mathcal{Y} = G_F(\mathcal{X})$ to denote a set of fuzzy rules, where $\mathcal{X}$ and $\mathcal{Y}$ are two fuzzy variables. Let the ith fuzzy rule from $\mathcal{X}$ to $\mathcal{Y}$ be given as "If $\mathcal{X} = w_i$ then $\mathcal{Y} = \nu_i$" $(i = 1, 2, ...)$. If we choose the TS fuzzy model structure, the ν_i can be a crisp

function of $\mathbf{x}$. In this chapter, functions which are affine with respect to $\mathbf{x}$ are used in local models as v_i, ie., $v_i = A_i\mathbf{x} + B_i$. Let $\mathbf{p}_i$ be the centres of w_i. Then $\mathcal{Y} = G_F(\mathcal{X})$ can be determined by $G_F(\mathbf{p}_i) = A_i\mathbf{x} + B_i$. This results in the affine TS fuzzy model (cf. (2.9) and (2.11))

$$\mathbf{y}_{t+1} = \sum_i w_i(\mathbf{x}_t)(A_i\mathbf{x}_t + B_i). \tag{3.5}$$

In light of (3.5), our objective of fuzzy modelling is to find both the membership functions $w_i(\cdot)$ and model parameters $A_i, B_i, (i = 1, 2, ...)$ such that the output of (3.5) approaches the real system output $\mathbf{y}_t$ asymptotically when excited by the same but arbitrary data sequence. Suppose there are c local models.

Denote

$$\Lambda(\mathbf{x}) = [\mathbf{x}^T w_1(\mathbf{x})\ w_1(\mathbf{x})\ \mathbf{x}^T w_2(\mathbf{x})\ w_2(\mathbf{x})\ ...\ \mathbf{x}^T w_c(\mathbf{x})\ w_c(\mathbf{x})], \tag{3.6}$$

and

$$Q = [A_1\ B_1\ A_2\ B_2\ ...\ A_c\ B_c]. \tag{3.7}$$

Using the definitions of Q and $\Lambda(\mathbf{x})$ in (3.6) and (3.7), equation (3.5) can be rewritten in the matrix form

$$\mathbf{y}_{t+1} = Q\Lambda^T(\mathbf{x}_t). \tag{3.8}$$

It is essentially a weighted sum of Gaussian functions. For any continuous nonlinear function $\mathbf{f}(\cdot)$ defined on a compact set D and an arbitrary small number ε, there always exist a matrix $\bar{Q}$ and a vector of Gaussian functions $\Lambda(\mathbf{x}_t)$, all of finite dimension, such that (Park and Sandberg, 1990)

$$\max_{\mathbf{x}\in D} \|\mathbf{f}(\mathbf{x}_t) - \bar{Q}\Lambda(\mathbf{x}_t)\| < \varepsilon. \tag{3.9}$$

This *universal approximation property* enables us to ascertain that (3.8) can indeed approximate any nonlinear function by adjusting the matrix Q and the parameters involved in the membership functions.

3.2 The modelling scheme

Suppose that a set of h sample data pairs denoted by $\{(\mathbf{x}_t, \mathbf{y}_t), 1 \le t \le h\}$ is obtained from an appropriate experiment. Our task now is to extract a fuzzy model of the form

$$\hat{\mathbf{y}}_{t+1} = Q\Lambda^T(\mathbf{x}_t). \tag{3.10}$$

from the data. The fuzzy modelling scheme is as follows:

1. Algorithm initialization: set $c = 1$. The membership function centre $\mathbf{p}_1$ is located on the centre of gravity of the input data, that is, the point whose components are the average values of corresponding components of all data. As there is only one multi-dimensional fuzzy sub-set covering the overall domain, the associated membership function $w_1(\mathbf{x}_t)$ is everywhere equal to 1. So $\Lambda(\mathbf{x}_t) = [\mathbf{x}_t^T \ 1]$ for any t, and our first model is

$$\hat{\mathbf{y}}_{t+1}^1 = Q\Lambda^T(\mathbf{x}_t). \tag{3.11}$$

where $Q = [A_1 \ B_1]$. Defining $\mathbf{e}_t^1 = \hat{\mathbf{y}}_t^1 - \mathbf{y}_t$ to be the modelling error at point t, Q in (3.11) can be estimated as follows:

$$\begin{aligned} Q_{t+1} &= Q_{t-1} - \frac{\alpha \mathbf{e}_t^1 \Lambda(\mathbf{x}_t)}{\beta + \Lambda(\mathbf{x}_t)\Lambda^T(\mathbf{x}_t)}, \ t = 1, \ldots h-1, \\ Q &= Q_h, \end{aligned}$$

where α and β are positive constants chosen as $0 < \alpha < 2$ and $\beta > 0$. Q_0 can be assigned arbitrarily.

2. Model expansion: At step k, define

$$\mathbf{e}_t^k = \hat{\mathbf{y}}_t^k - \mathbf{y}_t, t = 1, \ldots h. \tag{3.12}$$

Suppose $\mathbf{p}_k$ is the data point on which the modelling error is maximum in step $k-1$. Add $\mathbf{p}_k$ into the centre points set to form a new matrix $P = [\mathbf{p}_1 \ \mathbf{p}_2 \ \ldots \ \mathbf{p}_k]$. Define

$$\Delta_k = \min_{1 \le i,j \le k} \|\mathbf{p}_i - \mathbf{p}_j\|.$$

Determine σ_i such that

$$\frac{2}{3}\Delta_k \le \sigma_i \le \Delta_k, i = 1, 2, \ldots, k. \tag{3.13}$$

From (3.4), (3.5) and (3.6) we calculate $\Lambda(\mathbf{x}_t)$. The matrix Q in (3.10) is estimated as follows:

$$\begin{aligned} Q_{t+1} &= Q_{t-1} - \frac{\alpha \mathbf{e}_t^k \Lambda(\mathbf{x}_{t-1})}{\beta + \Lambda(\mathbf{x}_{t-1})\Lambda^T(\mathbf{x}_{t-1})}, \\ &\quad t = 1, \ldots h-1 \\ Q &= Q_h. \end{aligned} \tag{3.14}$$

Again, α and β are set same as in step 1.

3. Set $c = k$. If $\sum_t \|\mathbf{e}_t^k\|$ is small enough then stop, otherwise $k = k+1$, go to the above step.

Remark The α and β can be choosen by the designer within the permitted intervals. α will affect the speed of the convergence and a accuracy of the estimation, ie., the smaller the α, the slower of the convergence. the choice of σ_i in (3.13) is obtained on observation and experiment.

3.3 Convergence of the algorithm

The following theorem summarizes the convergence result of the proposed modelling algorithm:

Theorem 3.1 *Let $\mathbf{e}_t^k$ be defined as in (3.12). Choose α and beta so that $0 < \alpha < 2$, $\beta > 0$. There exists an integer k such that the modelling scheme described in preceding section guarantees that*

$$\lim_{t\to\infty} \|\mathbf{e}_t^k\| = 0.$$

Proof: In the following we use a standard method (cf. (Tan and Yu, 1996)) to prove the theorem. According to (3.9), there is a large enough integer k and a matrix $\bar{Q}$ whose number of column is k such that $\mathbf{y}_{t+1} = \bar{Q}\Lambda(\mathbf{x}_t)$. Subtracting $\bar{Q}$ from both sides of (3.14), and denoting $E_t = Q_t - \bar{Q}$, we have that

$$E_t - E_{t-2} = -\frac{\alpha \mathbf{e}_{t-1}^k \Lambda(\mathbf{x}_{t-2})}{\beta + \Lambda(\mathbf{x}_{t-2})\Lambda^T(\mathbf{x}_{t-2})}. \tag{3.15}$$

Define $V_t^i = E_t^i (E_t^i)^T$, with E_t^i being the ith row of the error matrix E_t $(i = 1, 2, ..., \eta)$. Then

$$V_t^i - V_{t-2}^i = (E_t^i - E_{t-2}^i)(E_t^i - E_{t-2}^i)^T + (E_t^i - E_{t-2}^i)(2E_{t-2}^i)^T. \tag{3.16}$$

Substituting (3.15) into (3.16) and noting that the ith row of $\mathbf{e}_t^k$ is $e_{ti}^k = E_{t-1}^i \Lambda^T(x_{t-1})$, we have

$$
\begin{aligned}
(E_t^i - E_{t-2}^i)(E_t^i - E_{t-2}^i)^T &= \frac{\alpha^2 (e^k_{(t-1)i})^2 \Lambda(\mathbf{x}_{t-2}) \Lambda^T(\mathbf{x}_{t-2})}{(\beta + \Lambda(\mathbf{x}_{t-2})\Lambda^T(\mathbf{x}_{t-2}))^2} \\
&\leq \frac{\alpha^2 (e^k_{(t-1)i})^2}{\beta + \Lambda(\mathbf{x}_{t-2})\Lambda^T(\mathbf{x}_{t-2})}, \\
(E_t^i - E_{t-2}^i)(2E_{t-2}^i)^T &= -\frac{2\alpha e^k_{(t-1)i} \Lambda(\mathbf{x}_{t-2}) (E_{t-2}^i)^T}{\beta + \Lambda(\mathbf{x}_{t-2})\Lambda^T(\mathbf{x}_{t-2})} \\
&= -\frac{2\alpha (e^k_{(t-1)i})^2}{\beta + \Lambda(\mathbf{x}_{t-2})\Lambda^T(\mathbf{x}_{t-2})}.
\end{aligned}
$$

Thus from equation (3.16) we have

$$
\begin{aligned}
V_t^i - V_{t-2}^i &\leq \frac{\alpha^2 (e^k_{(t-1)i})^2}{\beta + \Lambda(\mathbf{x}_{t-2})\Lambda^T(\mathbf{x}_{t-2})} - \frac{2\alpha (e^k_{(t-1)i})^2}{\beta + \Lambda(\mathbf{x}_{t-2})\Lambda^T(\mathbf{x}_{t-2})} \\
&= -\alpha(2-\alpha)\frac{(e^k_{(t-1)i})^2}{\beta + \Lambda(\mathbf{x}_{t-2})\Lambda^T(\mathbf{x}_{t-2})} \\
&\leq 0,
\end{aligned}
$$

ie,

$$
\alpha(2-\alpha)\frac{(e^k_{(t-1)i})^2}{\beta + \Lambda(\mathbf{x}_{t-2})\Lambda^T(\mathbf{x}_{t-2})} \leq V_{t-2}^i - V_t^i.
$$

Therefore,

$$
\begin{aligned}
\sum_{j=1}^{t} \alpha(2-\alpha)\frac{(e^k_{ji})^2}{\beta + \Lambda(\mathbf{x}_{j-1})\Lambda^T(\mathbf{x}_{j-1})} &\leq V_0^i + V_1^i - V_t^i - V_{t+1}^i \\
&\leq V_0^i + V_1^i,
\end{aligned}
$$

which leads to

$$
\lim_{t\to\infty} \sum_{j=1}^{t} \alpha(2-\alpha)\frac{(e^k_{ji})^2}{\beta + \Lambda(\mathbf{x}_{j-1})\Lambda^T(\mathbf{x}_{j-1})} \leq V_0^i + V_1^i.
$$

From the above we have that

$$
\lim_{t\to\infty} \frac{|e^k_{ti}|}{\sqrt{\beta + \Lambda(\mathbf{x}_{t-1})\Lambda^T(\mathbf{x}_{t-1})}} = 0
$$

for all i. But

$$
0 \leq \frac{|e^k_{ti}|}{\sqrt{\beta + n}} \leq \frac{|e^k_{ti}|}{\sqrt{\beta + \Lambda(\mathbf{x}_{t-1})\Lambda^T(\mathbf{x}_{t-1})}}.
$$

We conclude that $\lim_{t\to\infty} |e^k_{ti}| = 0$, and so

$$
\lim_{t\to\infty} ||\mathbf{e}_t^k|| = 0,
$$

proving the theorem.

3.4 Example

The well-known Box and Jenkins gas furnace data is used (Box and Jenkins, 1970) as an example to illustrate the above modelling algorithm. The data consist of 296 I/O measurements of a gas furnace system. The input $u(t)$ is the gas flow rate into the furnace and the output $y(t)$ is CO_2 concentration in outlet gas. The sampling interval is 9 seconds. The furnace's input and output data are shown in Figure 3.1 and Figure 3.2, respectively. We will now construct a fuzzy model of the form

$$y(t+1) = f(y(t), y(t-1), y(t-2), u(t), u(t-1), u(t-2)), \tag{3.17}$$

which was also suggested in Kim et al. (1997). Within our modelling framework, let

$$\begin{aligned} \mathbf{y}_t &= [y(t-2)\ y(t-1)\ y(t)]^T, \\ \mathbf{u}_t &= [u(t-2)\ u(t-1)\ u(t)]^T, \\ \mathbf{x}_t &= [\mathbf{y}_t^T\ \mathbf{u}_t^T], \end{aligned}$$

then we have

$$\mathbf{y}_{t+1} = F(\mathbf{x}_t),$$

which is a 6-input 3-output fuzzy model.

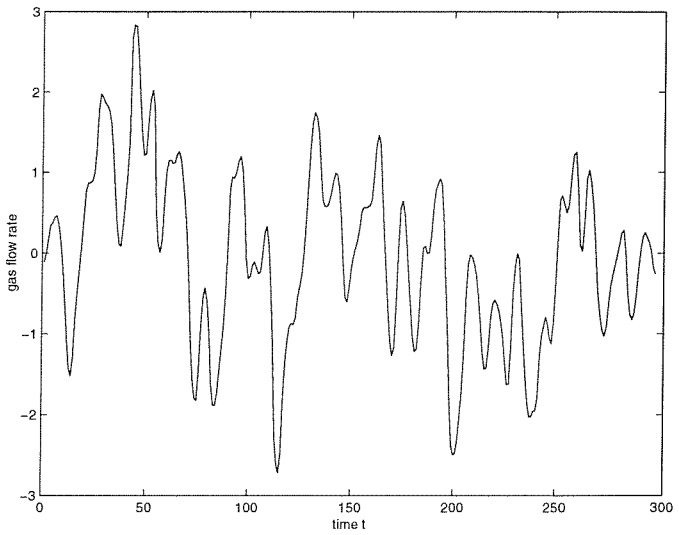

Figure 3.1: The input measurements of the gas furnace

Using the above proposed algorithm, we get a fuzzy model consisting of 4 rules which partition the 6-dimensional input space into 4 subspaces. The membership

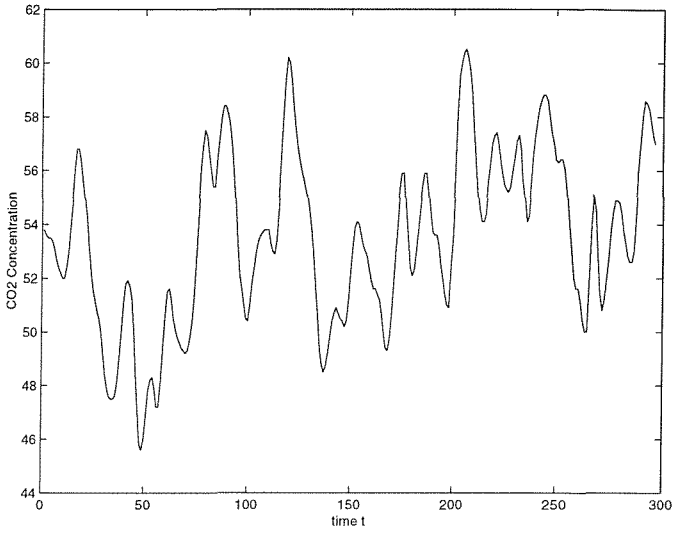

Figure 3.2: The output measurements of the gas furnace

Rule	$\mathbf{P}_i$	σ_i
1	[53.3901 53.4214 54.1673 -0.0173 -0.0470 0.2801]	3.5867
2	[53.6510 53.3827 53.1590 0.1840 -0.0374 -0.0346]	2.9874
3	[51.0111 52.3321 49.0901 1.5213 1.0112 1.8002]	1.2432
4	[56.2434 55.5421 55.0121 -1.8756 -1.3452 -1.1345]	2.0021

Table 3.1: The membership functions of Example 1

Rule	A_i	B_i
1	[0.4075 -1.4869 2.0160 -0.1521 -0.1636 0.0762]	3.4112
2	[-0.0913 -0.2034 1.0882 -1.5711 1.3619 -0.4190]	10.9817
3	[0 -0.5490 1.4131 -0.7552 0.4451 -0.1210]	7.3461
4	[0.3820 -1.3421 1.8589 -0.5358 0.1901 0.3817]	5.4141

Table 3.2: The local models of Example 1

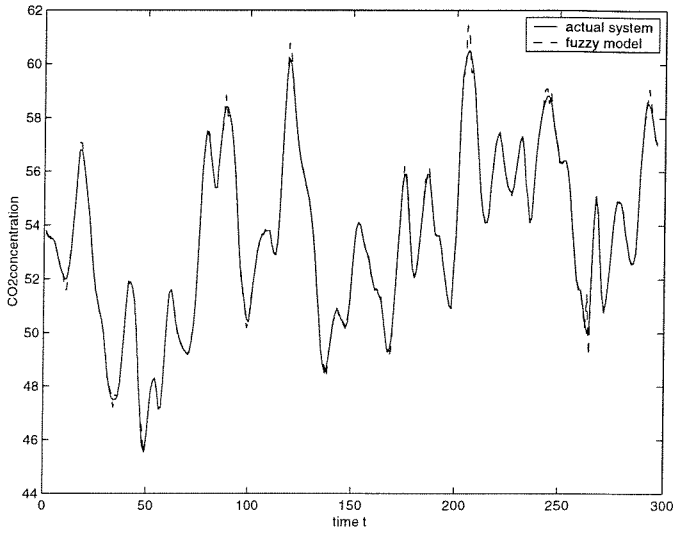

Figure 3.3: The modelling performance

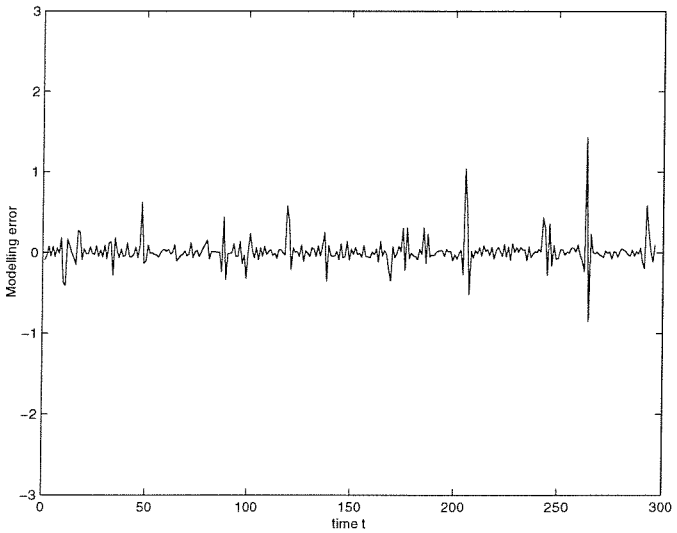

Figure 3.4: The modelling error

functions are shown in table 3.1 and the local models are shown in table 3.2 (only the last rows are shown). The modelling performance is shown in Figure 3.3. The modelling error is shown in Figure 3.4. We can see that even with such a simple structure the model is quite satisfactory. Note that if using the lattice partitioning algorithms such as that proposed in Tan and Yu (1996), then the number of rules will be substantially bigger due to the high dimension of the input space. For example, if each axis is partitioned into only two regions, then the total number of submodels would be $2^6 = 64$! Generally speaking, if the dimension of input vector varable x_t is n, if use lattice partitioning algorithms, then the number of local areas is an exponential function of n. But if use the proposed method, the number of local areas does not have to depend on n. Therefore it can avoid COD problem.

3.5 Optimal piecewise locally linear modelling

In the remainder of this chapter, we introduce an optimal piecewise locally linear modelling algorithm for nonlinear systems that is amenable to controller design, but utilise only input-output data to formulate the design (Harris, Hong and Feng, 1999).

In next section, a new *mixture of expert network* (MEN) (Harris, Hong and Feng, 1999) is introduced to form smooth piecewise linear models using a *Delaunay partition* of the input space (Sibson, 1978; Lo, 1989; Wu and Harris, 1998), a popular domain partition method in numerical analysis which is efficient in forming irregular regions. One of the main advantages of the MEN algorithm in dynamical system modelling is that the local models are formulated in a global form and consequently can be trained adaptively and on-line. At each iteration, each local linear model is selected to be trained using a Min/Max decision rule in the mixture of experts network. However the application of the method is restricted due to an *a priori* assumption of convexity of the output surface which determines the decision rules that are used to train the network. For a wider class of nonlinear systems with nonconvex multimodal output surfaces, global optimisation techniques may provide viable approaches to obtain the most appropriate Delaunay partitioned input space which is necessary to obtain smooth piecewise linear models with optimal approximation properties. In section 3.7. a *very fast simulated re-annealing* (VFSR) method is introduced for global optimisation of non-convex cost functions (Ingber, 1989; Ingber, 1992), which has been effectively applied in nonlinear signal processing applications (Chen and Luk, 1999). A new optimal piecewise local linear modelling algorithm is established based on the MEN and VFSR. The proposed algorithm consists of two training phases, global optimi-

sation of the input space partition using the VFSR procedure and a local linear model training using conventional gradient descent such as the *normalised least mean squares* (NLMS) method (Brown and Harris, 1994) with the introduction of a new mixture of experts network using a new decision rule derived from convex set theory. The VSFR method is applied with a global optimisation cost function to find an optimal set of vertices of a Delaunay triangulation that forms a collection of subsets of the input space in order to obtain an optimal piecewise linear model. The new MEN algorithm is formed with a new simple and fast decision rule for the gate control in training each local model, which is derived using convex set theory and is applicable to high dimensional Delaunay input space partitioning. In consequence the advantage of fast on-line properties of MEN is extended to unknown output surface convexity problems.

3.6 The decision rule for the training of piecewise linear model

The mixture of expert network is a hierarchical structure as shown in Figure 3.5 (Smith and Johnansen, 1997; Wu and Harris, 1998; Harris, Hong and Feng, 1999). The nodes

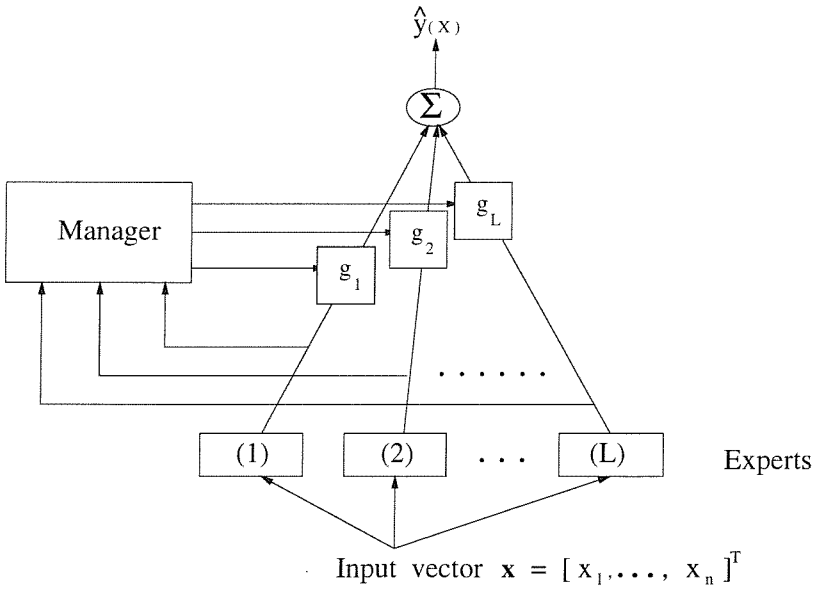

Figure 3.5: The mixture of experts network

of the first layer of the network represents the individual expert who sends a report to the manager. The manager monitors and evaluates all the reports to form a final opinion of the system output and how the network is to be trained, by operating the

gate g_i of each expert using some decision rules.

In the piecewise linear modelling problem, the experts correspond to a set of local linear models. Consider a system $y = f(\mathbf{x})$ where y denotes system output, $\mathbf{x}$ is system input vector with an appropriate dimension and $f(\cdot)$ is described by a piecewise linear model. Denote the input space S: $\mathbf{x} \in S \subset \Re^n$. A piecewise linear model consists of a set of L local linear models with a set of L operating ranges $S^{(1)}$, $S^{(2)}$, $\cdots$, $S^{(L)}$ which satisfy

$$S^{(1)} \bigcup S^{(2)} \bigcup \cdots \bigcup S^{(L)} = S, \tag{3.18}$$

and

$$S^{(j)} \bigcap S^{(k)} = \emptyset, \quad \text{when } j \neq k, \tag{3.19}$$

where $\emptyset$ denotes empty set. Each local model is written as

$$\hat{y}_i(\mathbf{x}) = [1 \quad \mathbf{x}^T]\hat{\Theta}^{(i)}, \text{ IF } \mathbf{x} \in S^{(i)} \text{ , } i = 1, \cdots, L, \tag{3.20}$$

where $\hat{\Theta}^{(i)} \in \Re^{n+1}$ denotes parameter estimate vector used in ith local linear model.

The output of the network is defined by

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{L} g_i \hat{y}_i(\mathbf{x}) \tag{3.21}$$

with

$$g_i = \begin{cases} 1 & \text{IF } \mathbf{x} \in S^{(i)} \\ 0 & \text{otherwise.} \end{cases}$$

For the application of mixture of experts network to dynamical system modelling and time series that the training of the network is carried out in an adaptive and on-line processing manner, some fast decision rules for the control of the gate g_i is desirable. Before developing such a rule, firstly we need to introduce the concept of *Delaunay triangulation* (Sibson, 1978; Lo, 1989). For simplicity, we only discuss the two dimensional case. For Delaunay triangulation in a n-dimensional Euclidean space ($n > 2$), please refer to Brostow and Dussault (1978) or Bowyer (1981).

Let $\mathbf{P} = \{P_i, i = 1, ..., N\}$ be a set of N points in $\Re^2$, and define the set of polygons $\mathbf{V} = \{V_i, i = 1, ..., N\}$ where $V_i = \{\mathbf{x} \in \Re^2 | \ \|\mathbf{x} - P_i\| < \|\mathbf{x} - P_j\|, \forall j \neq i\}$, and $\|\cdot\|$ denotes Euclidean distance norm. V_i represents a region of $\Re^2$ whose points are nearer to node point P_i than to any other points. Thus, V_i is an open convex polygon, usually called a *Voronoi polygon* (Brostow and Dussault, 1978), whose boundaries

are portions of the perpendicular bisectors of the lines joining node P_i to node P_j when V_i and V_j are contiguous. The collection of Voronoi polygons $\mathbf{V}$ is called the *Dirichlet tessellation* (Bowyer, 1981). In general, a vertex of a Voronoi polygon is shared by two other neighbouring polygons so that connecting the three node points associated with such adjacent polygons forms a triangle, say, T_k. The set of triangles $\{T_k\}$ is called the *Delaunay triangulation*. This construction can be shown to be a triangulation of a convex hull for the set of node points $\mathbf{P}$ (Sibson, 1978). Figure 3.6 gives a demonstration of these concepts for $n = 2$ and $N = 5$.

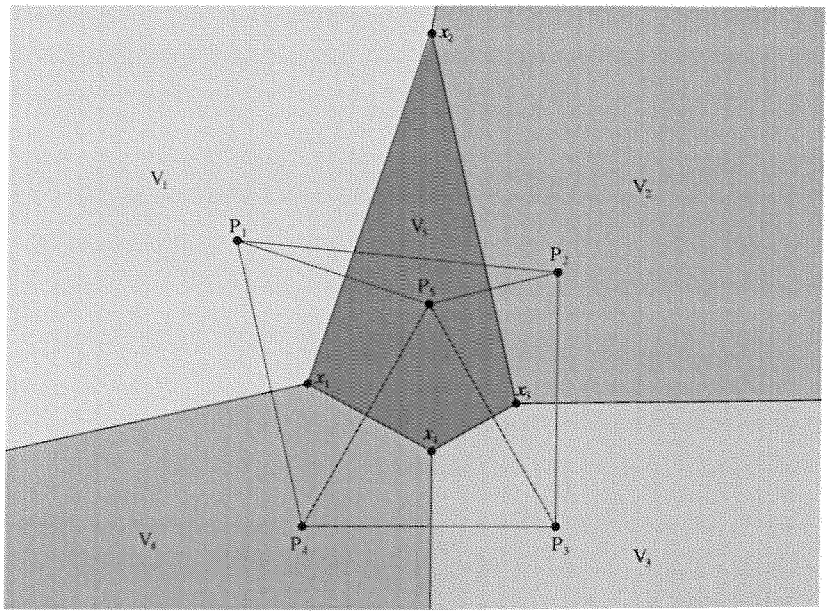

Figure 3.6: A Delaunay triangulation in $\Re^2$: Node points: $\mathbf{P} = \{P_1, P_2, P_3, P_4, P_5\}$, Voronoi polygons: $\{V_1, V_2, V_3, V_4, V_5\}$, vertices of Voronoi polygons: $\{x_1, x_2, x_3, x_4\}$, Delaunay triangulation of $\mathbf{P}$: $\{\triangle P_1P_4P_5, \triangle P_1P_2P_5, \triangle P_2P_3P_5, \triangle P_3P_4P_5\}$

An important property of the Delaunay triangulation is that any three node points will form a Delaunay triangle if and only if the circumcircle defined by these three nodes contains no other node points in its interior. Note that a Delaunay triangle is a triangle in 2D, a tetrahedron in 3D, and so on. Each Delaunay triangle connects $n+1$ vertices in the n-dimensional space. Although Delaunay triangulation is a useful tool in *finite element method*, its application has primarily been to convex hulls. In the sequel, Delaunay triangulations are used to develop a fast decision rule for the control of the gate g_i.

Consider the model input space operating ranges $\{S^{(i)}\}$ which is formed using Delaunay triangles as $S^{(i)} = \mathcal{H}\{\mathbf{v}_0^{(i)}, \mathbf{v}_1^{(i)}, \cdots, \mathbf{v}_n^{(i)}\}$, where $\{\mathbf{v}_0^{(i)}, \mathbf{v}_1^{(i)}, \cdots, \mathbf{v}_n^{(i)}\}$ denote the vertices of $\{S^{(i)}\}$, by allowing some points freely within or on the hull of the

input space to form these vertices, as illustrated in Figure 3.7. The free parameters determining the positions of these free points can be written as a vector $\omega \in \Re^p$ with an appropriate dimension p, which can be optimally determined using the VFSR algorithm described in next section. Note that with present method, the transformation from ω to $\{\mathbf{v}_0^{(i)}, \mathbf{v}_1^{(i)}, \cdots, \mathbf{v}_n^{(i)}\}$ must be explicitly defined by the modeller subject to the neighbour information produced when he or she defines and labels these Delaunay triangulations satisfying (3.18) and (3.19). These neighbour information involve a few constraints, such as, some of the vertices use the natural vertices of the input domain, some of the free point are constrained on the hull of input domain, and the vertices of different Delaunay triangles use a coincident point. As a consequence of these constraints, the dimension size of free parameter ω can be reduced to a small number. For example, the input partition of Figure 3.7 involves only 8 free parameters. Hence the derivation of $\{\mathbf{v}_0^{(i)}, \mathbf{v}_1^{(i)}, \cdots, \mathbf{v}_n^{(i)}\}$ from ω would not incur much difficulty because it is always advised to use as small number of free points as possible in order to overcome the curse of dimensionality. For this purpose it is also recommended that preprocessing of variable selection is used to reduce n, the dimension of input vector $\mathbf{x}$, prior to modelling for very complicated systems using conventional signal processing algorithms such as projection pursuit, principal component analysis (PCA), canonical analysis or some special model construction algorithm (Bossley, 1997; Hong, Harris and Wilson, 1998; Huang, Harris and Nixon, 1998).

It can be shown from convex set theory that any point $\mathbf{x}$ within $S^{(i)}$ is a unique convex combination of $\{\mathbf{v}_0^{(i)}, \mathbf{v}_1^{(i)}, \cdots, \mathbf{v}_n^{(i)}\}$ as (Stoer and Witzgall, 1970)

$$\mathbf{x} = \sum_{j=0}^{n} \lambda_j^{(i)} \mathbf{v}_j^{(i)}, \tag{3.22}$$

where $\lambda_j^{(i)} \geq 0$, $\sum_{j=0}^{n} \lambda_j^{(i)} = 1$. Denote $\Lambda^{(i)} = [\lambda_1^{(i)}, \cdots, \lambda_n^{(i)}]^T \in \Re^n$ and substitute $\lambda_0^{(i)} = 1 - \sum_{j=1}^{n} \lambda_j^{(i)}$ into (3.22), yielding

$$\Lambda^{(i)} = \left[\tilde{V}^{(i)}\right]^{-1} (\mathbf{x} - \mathbf{v}_0^{(i)}), \tag{3.23}$$

where $\tilde{V}^{(i)} = [\mathbf{v}_1^{(i)} - \mathbf{v}_0^{(i)}, \cdots, \mathbf{v}_n^{(i)} - \mathbf{v}_0^{(i)}] \in \Re^{n \times n}$. Hence the IF statement in (3.20) can be represented through L experts opinions given by (3.23). The manager chooses the gate g_i by using a new decision rule, that is, to check the solutions given by L set of $\Lambda^{(i)}$, $i = 1, \cdots, L$ and to select that which satisfies the constraints $\lambda_j^{(i)} \in [0, 1)$ for $1 \leq j \leq n$ and $\sum_{j=1}^{n} \lambda_j^{(i)} \in [0, 1)$, and then let $g_i = 1$.

The training of local linear models can be carried out adaptively in a on-line manner within a global framework using the mixture of experts network, and is summarized

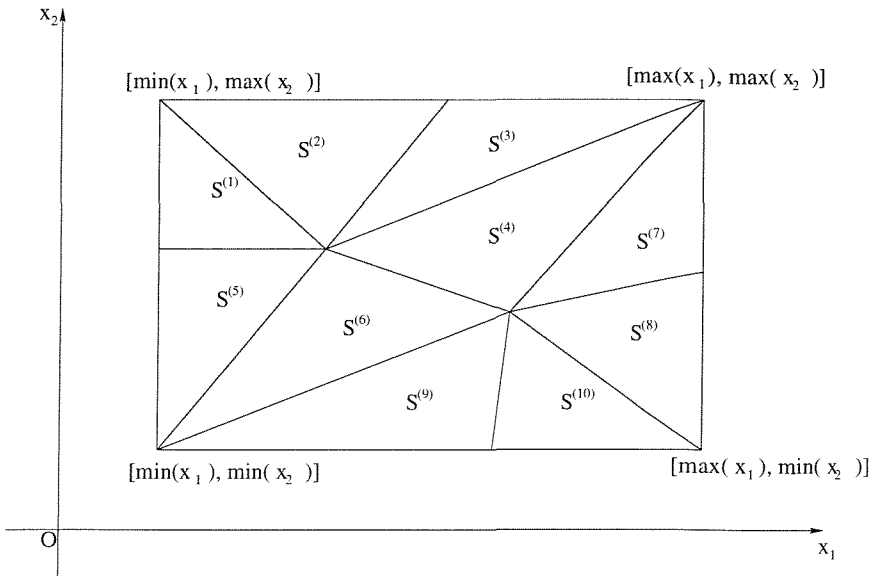

Figure 3.7: Delaunay triangulation partition of input space

as follows.

1. Determine the input space domain (the convex set S) constrained within the maximum and minimum value of the data range of the input vector $\mathbf{x}$ by using data preprocessing. Given an appropriate number of points which located within or on the hull of the input space to form vertices of a set of L subranges (simplexes, Delaunay triangles), denoted as $S^{(i)} = \mathcal{H}\{\mathbf{v}_0^{(i)}, \mathbf{v}_1^{(i)}, \cdots, \mathbf{v}_n^{(i)}\}$. Corresponding to each $S^{(i)}$, given a set of initial parameters for L local linear models $\hat{\Theta}^{(i)} \in \Re^{n+1}$, $i = 1, 2, \cdots, L$.

2. At iteration t, the training sample is given as $\{y(t), \mathbf{x}(t)\}$. Each expert opinion is formed using the solution of (3.23) (where $\mathbf{x}$ is replaced by $\mathbf{x}(t)$) and reported to the manager. The manager selects the ith model that satisfies the constraints $\lambda_j^{(i)} \in [0, 1)$ for $1 \leq j \leq n$ and $\sum_{j=1}^{n} \lambda_j^{(i)} \in [0, 1)$, and then sets $g_i = 1$, $g_k = 0$ for $k \neq i$.

3. The output of the network is produced and the network weights is adjusted using the NLMS algorithm as

$$\begin{aligned} \hat{y}(t) &= [1 \quad \mathbf{x}(t)^T]\hat{\Theta}^{(i)}(t-1), \\ \hat{\Theta}^{(i)}(t) &= \hat{\Theta}^{(i)}(t-1) + \eta \frac{[1 \quad \mathbf{x}^T(t-1)]^T e(t)}{c + [1 \quad \mathbf{x}^T(t-1)][1 \quad \mathbf{x}^T(t-1)]^T}, \\ e(t) &= y(t) - \hat{y}(t), \end{aligned} \tag{3.24}$$

where η is a preset learning rate, c is an arbitrarily small positive constant.

4. The overall performance of the network can be measured using the mean square error (MSE) as $J = \sum_{t=1}^{N} e^2(t)$. The network output is a highly complex nonlinear

function of the positions of the vertices which determines how the input space is partitioned. The optimization of J with respect to the vertices position to generate the optimal piecewise linear model can be realized using the VFSR algorithm given in the following section.

3.7 VFSR for optimal Delaunay input space partition

simulated annealing (SA) belongs to a class of so called *guided random search* methods. It evolves a single solution in the parameter space with certain guiding principles that imitate the random behaviour of molecules during the annealing process, ie., it represents a global optimization technique with some striking positive and negative features. An attractive feature of SA is that it is very easy to program and the algorithm typically has few parameters that require tuning. An elegant discussion on how the general SA algorithm works can be found in Rosen (1997).

However, a serious drawback of SA is that it is often very slow. The VFSR algorithm is a global optimization method which is modified from the traditional SA algorithm to overcome the disadvantage of the slow convergence of the SA but maintains the advantage of the simplicity of SA. It employ a very fast annealing schedule, as it has self adaptation ability to re-scale temperatures. The VFSR is very suitable for constrained optimization problems, such as a search for an optimal parameter vector of limited dimension within a constraint space. In the present problem the free parameter vector ω forming the free vertices of the Delaunay triangulation is the parameter vector chosen to be optimized. One implementation of VFSR algorithm, described in Chen and Luk (1999) or Rosen (1997) will be used in this work. The proposed algorithm is as follows:

1. In the initialisation, an initial $\omega \in \Re^p$ is randomly generated within a feasible set as $\Omega : L_i \leq \omega_i \leq H_i$. The values of L_i and H_i correspond to some $\min[x_j]$ and $\max[x_j]$ respectively. the initial temperature of the acceptance probability function, $T_c(0)$, is set to $J(\omega)$, and the initial temperatures of the parameter generating probability functions, $T_i(0), 1 \leq i \leq p$, are set to 1.0. A user defined control parameter c in annealing is given, and the annealing times k_i for $1 \leq i \leq p$ and k_c, are all set to 0.

2. The algorithm generates a new point in the parameter space with:

$$\omega_i^{new} = \omega_i^{old} + q_i(H_i - L_i), \quad \text{for } 1 \leq i \leq p \text{ and } \omega_i^{new} \in \Omega, \tag{3.25}$$

where q_i is calculated as

$$q_i = \text{sgn}(v_i - 0.5)T_i(k_i)\left((1+\frac{1}{T_i(k_i)})^{|2v_i-1|}-1\right), \tag{3.26}$$

and v_i is a uniformly distributed random variable in [0,1]. If ω_i^{new} is not in Ω it is discarded until a new $\omega_i^{new} \in \Omega$ is produced. The ω_i^{new} is then transformed to $\{\mathbf{v}_0^{(i)}, \mathbf{v}_1^{(i)}, \cdots, \mathbf{v}_n^{(i)}\}$ by the user defined labelling. The local linear models algorithm given in the last section is used to obtain a new $J(\omega^{new})$ which is then evaluated and the acceptance probability of ω^{new} is given by

$$P_{accept} = \frac{1}{1+\exp[(J(\omega^{new})-J(\omega^{old}))/T_c(k_c)]}. \tag{3.27}$$

A uniform random variable P_{unif} is generated in [0,1]. If $P_{unif} \leq P_{accept}$, ω^{new} is accepted; otherwise it is rejected.

3. Denote ω^{best} as the best point amongst a predetermined number N_{accept} acceptance points. The reannealing procedure involves first calculating the sensitivities

$$s_i = |\frac{J(\omega^{best}+\mathbf{e}_i\delta)-J(\omega^{best})}{\delta}|, \text{ for } 1 \leq i \leq p, \tag{3.28}$$

where δ is a small step size, and $\mathbf{e}_i \in \Re^p$ is a unit vector with ith element as 1, other elements as 0. Let $s_{max} = \max\{s_i, 1 \leq i \leq p\}$. Each parameter generating temperature T_i is scaled by a factor s_{max}/s_i and the annealing time k_i is reset

$$T_i(k_i) = \frac{s_{max}}{s_i}T_i(k_i), \qquad k_i = \left(-\frac{1}{c}\log[\frac{T_i(k_i)}{T_i(0)}]\right)^p. \tag{3.29}$$

Similarly, $T_c(0)$ is reset to the value of the last accepted cost cost function, $T_c(k_c)$ is reset to $J(\omega^{best})$ and the annealing time k_c is rescaled accordingly

$$k_c = \left(-\frac{1}{c}\log[\frac{T_c(k_c)}{T_c(0)}]\right)^p. \tag{3.30}$$

4. After every N_{genera} generated points, annealing takes places with

$$\begin{aligned} k_i &= k_i+1 \\ T_i(k_i) &= T_i(0)\exp(-ck_i^{\frac{1}{p}}), \end{aligned} \qquad 1 \leq i \leq p$$

and

$$\begin{aligned} k_c &= k_c+1 \\ T_c(k_i) &= T_c(0)\exp(-ck_c^{\frac{1}{p}}). \end{aligned} \qquad 1 \leq i \leq p$$

Otherwise goto step 2.

5. The algorithm is terminated if the parameters has remained unchanged for a few successive reannealings or a preset maximum number of cost function evaluations has been reached; Otherwise, goto step (2).

3.8 Numerical example

Consider the benchmark *nonlinear autoregressive* (NAR) time series (Chen and Billings, 1992)

$$\begin{aligned} y(t) &= (0.8-0.5\exp(-y^2(t-1)))y(t-1)-(0.3+0.9\exp(-y^2(t-1)))y(t-2) \\ &\quad +0.1\sin(\pi\times y(t-1))+e(t), \end{aligned} \tag{3.31}$$

where π is the *ratio of the circumference of a circle to its diameter*, the noise $e(t)$ is a *Gaussian white sequence* with mean zero and variance 0.02. 1000 data points were generated and the first 500 points were used as an estimation data set. The remaining data were used as a validation data set. The input vector is set as $\mathbf{x}=[y(t-1),y(t-2)]^T$. The input space partition is shown in Figure 3.8, where the rectangle *ABCD* represents the bounds of a slightly enlarged input space S. 4 points *E*,*F*,*G*,*H* are allowed to move along each side of the rectangular *ABCD* and point *K* is a free point within the input space, forming $L=8$ triangles $S^{(i)}$'s that can be determined using a free parameter vector ω of dimension size $p=6$. The VFSR algorithm was applied to determine the parameter vector ω, or the position of points E, F, G, H and K, as plotted in Figure 3.8. At this identification stage, the local linear models were trained using ordinary least squares to reduce computation time. 8 local linear models of (3.20) were obtained as

$$\begin{aligned} \hat{y}_1(\mathbf{x}(t)) &= -0.0605+0.6845y(t-1)-0.6636y(t-2), \text{ IF } \mathbf{x}(t)\in S^{(1)}, \\ \hat{y}_2(\mathbf{x}(t)) &= -0.4503+1.1478y(t-1)-0.7507y(t-2), \text{ IF } \mathbf{x}(t)\in S^{(2)}, \\ \hat{y}_3(\mathbf{x}(t)) &= -0.4463+1.3563y(t-1)-0.9220y(t-2), \text{ IF } \mathbf{x}(t)\in S^{(3)}, \\ \hat{y}_4(\mathbf{x}(t)) &= -0.1022+0.9262y(t-1)-1.1197y(t-2), \text{ IF } \mathbf{x}(t)\in S^{(4)}, \\ \hat{y}_5(\mathbf{x}(t)) &= -0.1336+0.2907y(t-1)-1.0535y(t-2), \text{ IF } \mathbf{x}(t)\in S^{(5)}, \\ \hat{y}_6(\mathbf{x}(t)) &= 0.2329+0.8593y(t-1)-0.6843y(t-2), \text{ IF } \mathbf{x}(t)\in S^{(6)}, \\ \hat{y}_7(\mathbf{x}(t)) &= 0.2200+1.2104y(t-1)-1.0669y(t-2), \text{ IF } \mathbf{x}(t)\in S^{(7)}, \\ \hat{y}_8(\mathbf{x}(t)) &= 0.1853+0.2381y(t-1)-1.0057y(t-2), \text{ IF } \mathbf{x}(t)\in S^{(8)}, \end{aligned} \tag{3.32}$$

The results of one-step ahead prediction using these local linear models are plotted in Figure 3.8 demonstrating the good predictive performance of the amended MEN model. The VFSR algorithm in this application is still slow, however once an appropriate input partition is obtained, the on-line processing using the proposed algorithm can be carried out successfully. The model validity test and the one-step ahead prediction over the validation data set $t=800\sim900$ are plotted in Figure 3.9 demonstrating the model (3.32) is appropriate. The MSE of the validation data set $t=501\sim1000$ is $(0.023)^2$.

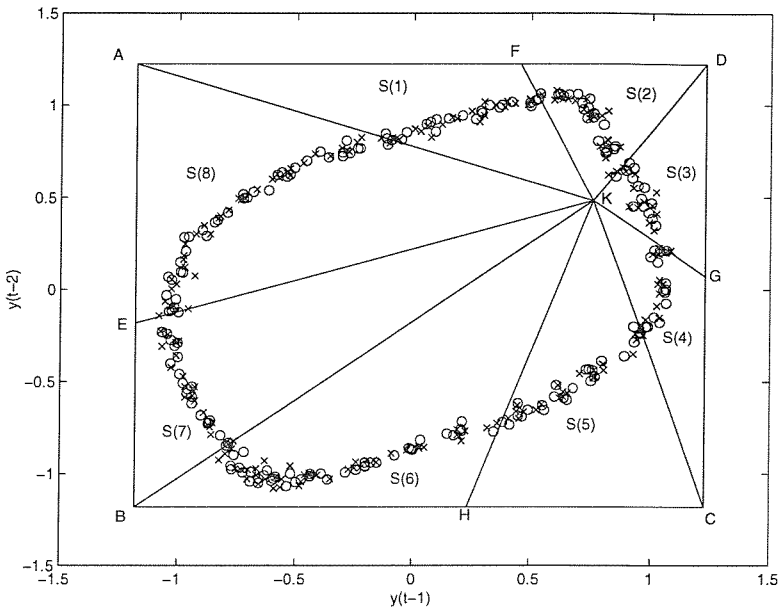

Figure 3.8: Piecewise locally linear modelling for numerical example (Circles: System measurements and Xs: one-step ahead predictions)

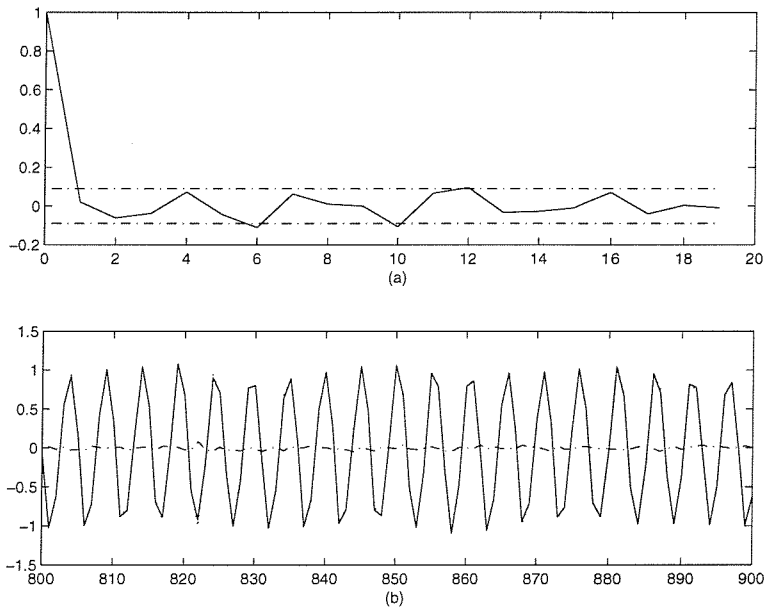

Figure 3.9: Model validation for numerical example: (a) Model validity test $\Phi_{ee}(\tau)$ and (b)one-step ahead prediction over the validation data set (Solid Line: System measurements, Dotted Line: one-step ahead prediction and Dash-dot Line: model residual)

In summary, the algorithm is efficient for the construction of Delaunay triangulation input space partitioned optimal piecewise locally linear model. Within the framework of a mixture of experts network, a new fast decision rule derived using convex set theory has been developed to train the model. A very fast simulated reannealing (VFSR) algorithm is utilised to search a global optimal solution for the Delaunay triangulation input space partition. The new approach can partly overcome the problem of discontinuity of lattice partitioned piecewise local linear models.

3.9 Comparision of two methods

To compare the two methods presented in this chapter, we apply the modelling scheme presented in section 3.2 to model system (3.31. We aim at constructing a fuzzy model of the form

$$y(t+1) = f(y(t), y(t-1)). \tag{3.33}$$

Let $\mathbf{y}_t = [y(t)\ y(t-1)]^T$ and $\mathbf{x}_t = \mathbf{y}_t$, then (3.33) can be written as a 2-input 2-output fuzzy model:

$$\mathbf{y}_{t+1} = \sum_i w_i(\mathbf{x}_t)[A_i\mathbf{x}_t + B_i]. \tag{3.34}$$

Using the algorithm presented in section 3.2, the modelling procedure can stop at $i = 4$, the last rows of obtained A_i, B_i and parameters of w_i are given in table 3.3. The system measurements, one-step ahead predictions and modelling errors are given in figure 3.10.

Rule	A_i	B_i	$\mathbf{P}_i$	σ_i
1	[1.2503 -0.8506]	-0.4432	[0.7880 0.6532]	0.5200
2	[0.4611 -0.8340]	0.0625	[-0.2374 0.6846]	0.9874
3	[1.0350 -0.8740]	0.2280	[-0.7681 -0.8786]	0.4432
4	[0.6085 -1.0901]	-0.1200	[0.2452 -0.6345]	0.7021

Table 3.3: The T-S fuzzy model of Example 2

Comparing the preceeding results with the simulation results in last section we can see that:

- There is no remarkable difference between the modelling capabilities of the two methods.

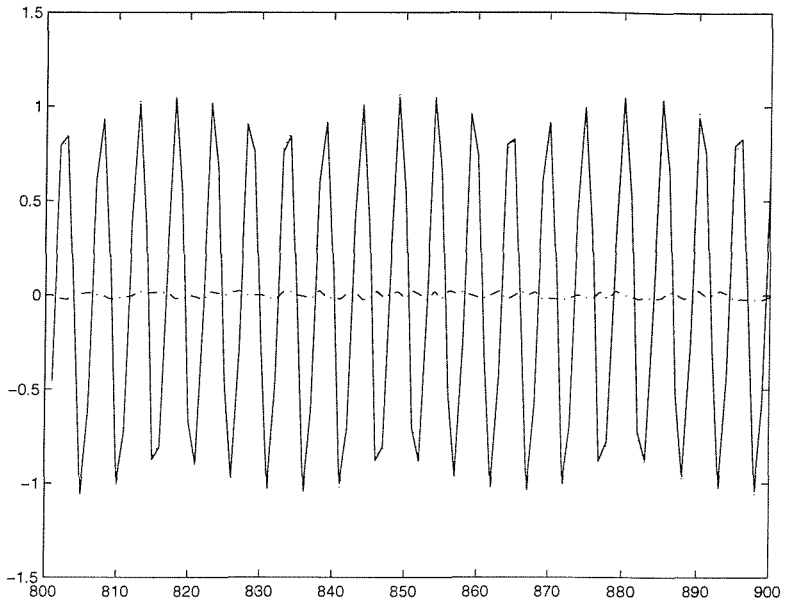

Figure 3.10: Fuzzy modelling for example 2 (Solid line: System measurements. Dotted line: one-step ahead predictions, and Dash-dot line: model residual)

- The first scheme is able to produce less number of local models than the second one. This is because the first method has included membership functions.
- The modelling speed of the first scheme is much faster than the second one because the VFSR algorithm in this application is still slow.
- The form of the final model of the second method is simpler than that of the first one because the second model only involves local linear models whereas the first one needs to take account of membership functions.

3.10 Concluding remarks

In this chapter, two new modelling algorithms have been introduced for constructing models of nonlinear systems from data. The new approaches can partly overcome the problem of discontinuity of lattice partitioned piecewise local linear models and the problem of COD since we partition the input space that is not based upon a lattice structure.

The first approach partitions the operating point space in a fuzzy clustering manner but avoids the requirement of *a priori* known number of clusters by a kind of mechanism which can *auto-select* the new cluster centre. In this way, it is able to derive

fuzzy models from data automatically and avoid the problems mentioned above.

Although the second method is introduced in quite a general framework, users are advised to apply the method to relatively low dimension problems due to the irregular shape of input space partitions, or to use input variable selection preprocessing.

Whilst the VFSR algorithm is slow in finding the optimal input space partition, this does not change the effectiveness of the on-line algorithm described in this chapter if an appropriate input partition had already been found. The new algorithm may also be combined with other global optimization approaches such as genetic algorithms, but these are likely to be even slower than VFSR.

Finally, the principles of the proposed techniques can also be used in other kinds for local modelling, which is an open research topic.

Chapter 4

Local stability

Stability is one of the most fundamental properties of dynamic systems, and many concepts have been introduced for the mathematical study of stability. The stability methods and results from the general nonlinear systems theory also apply to the systems considered in this thesis which are special cases of nonlinear systems. Among various stability theories, Lyapunov techniques are very useful in system analysis. Not only do they allow stability analysis and gain computation, but they are also useful in the solution of optimal control problems. This makes Lyapunov techniques a natural basis for analysis of local linear systems. The main obstacle to a direct application of the existing techniques is the nontrivial step of finding the appropriate Lyapunov function. Hence, methods for efficient Lyapunov function construction are of fundamental importance in a useful theory for local modelling and control.

In this section we give the basic Lyapunov stability results relevant to local modelling and control topics. In section 4.1, some basic concepts of Lyapunov stability are presented. Quadratic and piecewise Lyapunov stability functions for continuous time systems are discussed in section 4.2 and section 4.3, respectively. The corresponding results for discrete time systems are presented in section 4.4. Section 4.5 gives a number of useful results concerning linear matrix inequalities which will be used in subsequent chapters for developing new Lyapunov stability results for local systems.

4.1 Lyapunov stability

Primarily Lyapunov stability concerns autonomous systems of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{4.1}$$

with $\mathbf{x} = \mathbf{0}$ as an equilibrium point. The following definition (Khalil, 1992) makes more precise what is meant by stability in the sense of Lyapunov for a equilibrium point of an autonomous system. The state $\mathbf{x}_0 = \mathbf{x}(0)$.

Definition 4.1 *The equilibrium point* $\mathbf{x} = \mathbf{0}$ *of (4.1) is*

- ***stable*** *if, for each* $R > 0$*, there exists* $r = r(R) > 0$ *such that*

$$\|\mathbf{x}_0\| < r \implies \|\mathbf{x}(t)\| < R, \ \forall t \geq 0,$$

- ***unstable*** *if not stable,*

- ***asymptotically stable*** *if it is stable and r can be chosen such that*

$$\|\mathbf{x}_0\| < r \implies \lim_{t \to \infty} \mathbf{x}(t) = 0,$$

- ***exponentially stable*** *if, for each* $R > 0$*, there exists* $r = r(R) > 0$ *and two numbers* $c_1 = c_1(R) > 0$ *and* $c_2 = c_2(R) > 0$ *such that*

$$\|\mathbf{x}_0\| < r \implies \|\mathbf{x}(t)\| < c_1 e^{-c_2 t} \|\mathbf{x}_0\|, \ \forall t \geq 0.$$

Since $\|\mathbf{x}(t)\| \to 0$ when $t \to \infty$ for an exponentially stable equilibrium point, exponential stability implies asymptotic stability. However, the converse is not true, as can be shown by constructing examples which are stable but converge slower than any exponential function $e^{-c_2 t}$ (Slotine and Li, 1991).

The geometrical implication of stability, instability, asymptotic stability and exponential stability are shown in Figure 4.1.

Definition 4.1 is formulated to characterize the *local* behaviour of system trajectories starting near the equilibrium point. Local properties say nothing about the trajectories when starting some distance away from the equilibrium point. For this reason, *global* concepts about stability are required.

Definition 4.2 *The equilibrium point* $\mathbf{x} = \mathbf{0}$ *of (4.1) is*

- ***globally asymptotically stable*** *if it is asymptotically stable for any initial states.*

- ***globally exponentially stable*** *if it is exponentially stable for any initial states.*

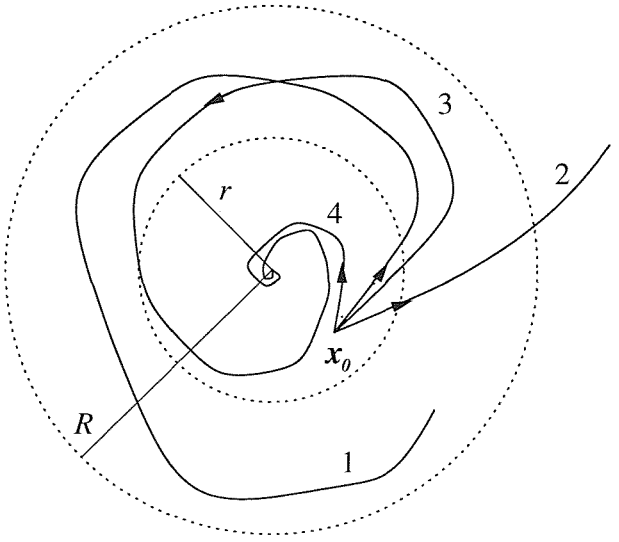

Figure 4.1: The concepts of stability

It is easy to see from the definition 4.2 that if the equilibrium point $\mathbf{x} = \mathbf{0}$ of a system is globally exponentially (asymptotically) stable, then it must be the unique equilibrium point of the system.

Based on the definitions 4.1 and 4.2, the Lyapunov stability theorem is stated in the following for the continuous time case (Khalil, 1992):

Theorem 4.1 *(Lyapunov's stability theorem—continuous time systems) Let* $\mathbf{x} = \mathbf{0}$ *be an equilibrium point for (4.1) and let* $V : D \longrightarrow R$ *be a continuously differentiable scalar function on a neighborhood* D *of* $\mathbf{x} = \mathbf{0}$ *such that*

$$V(\mathbf{0}) = 0 \text{ and } V(\mathbf{x}) > 0 \text{ in } D - \{\mathbf{0}\}, \tag{4.2}$$

$$\dot{V}(\mathbf{x}) \leq 0 \text{ in } D. \tag{4.3}$$

Then, $\mathbf{x} = \mathbf{0}$ *is stable. Moreover, if*

$$\dot{V}(\mathbf{x}) < 0 \text{ in } D, \tag{4.4}$$

then $\mathbf{x} = \mathbf{0}$ *is asymptotically stable in* D.

The function $V(\mathbf{x})$ is called a *Lyapunov function.* The Lyapnov function can sometimes be interpreted as an energy function. If the energy of the system is always positive and decreases along every trajectory of the system then the trajectory eventually will approach the origin.

If in addition to the conditions in theorem 4.1, also the condition

$$V(\mathbf{x}) \longrightarrow \infty \text{ for } \|\mathbf{x}\| \longrightarrow \infty \tag{4.5}$$

is satisfied, then $\mathbf{x} = \mathbf{0}$ is globally asymptotically stable. A function satisfying condition (4.5) is said to be *radially unbounded*. The notation $\|\cdot\|$ in this chapter stands for an arbitrary vector norm in $\Re^n$, for instance, the Euclidean norm (l_2 norm) $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ or the sup norm (l_∞ norm) $\|\mathbf{x}\| = \max\{|x_1|, ..., |x_n|\}$. Any two norms in a finite-dimensional vector space are equivalent in the sense that they define the same convergence (Chillingworth, 1976).

The conditions in Lyapunov's stability theorem are only sufficient. Failure of a Lyapunov function candidate to fulfill the conditions does not mean that the equilibrium is not stable or asymptotically stable. It simply means that the current candidate does not suffice.

A Lyapunov function can have arbitrary structure. In the local system modelling and control area it is, however, common to mainly consider the following types of Lyapunov functions:

- quadratic Lyapunov functions
- piecewise quadratic Lyapunov functions
- Piecewise affine Lyapunov functions

For our interest in the following we describe the first two types.

4.2 Quadratic Lyapunov functions

Quadratic forms is a class of functions for which it is easy to verify sign definiteness, ie., the case

$$V(\mathbf{x}) = \mathbf{x}^T P \mathbf{x}, \tag{4.6}$$

where P is a real symmetric matrix. In this case, $V(\mathbf{x})$ is *positive definite* (*semi positive definite*) if and only if all the eigenvalues of P are positive (nonnegative). If $V(\mathbf{x}) = \mathbf{x}^T P \mathbf{x}$ is positive definite (semi positive definite), we say that the matrix P is *positive definite* (*semi positive definite*) and write $P > 0$ ($P \geq 0$).

For a linear time-invariant autonomous system

$$\dot{\mathbf{x}} = A\mathbf{x}$$

the derivative of $V(\mathbf{x})$ along the trajectories of the system is given by

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \mathbf{x}^T P \dot{\mathbf{x}} + \dot{\mathbf{x}}^T P \mathbf{x} \\ &= \mathbf{x}^T (PA + A^T P)\mathbf{x} \\ &= -\mathbf{x}^T Q \mathbf{x}. \end{aligned}$$

Hence, in the linear case the quadratic Lyapunov stability test involves finding a matrix P that fulfills the two conditions

$$\begin{aligned} P &> 0, \\ PA + A^T P &< 0. \end{aligned} \tag{4.7}$$

The inequality (4.7) is called *Lyapunov inequality*. The solution to this can be obtained analytically by solving the *Lyapunov equation* $PA + A^T P = -Q$. A solution exists and is unique if and only if A is a *stable matrix* (a *Hurwitz matrix*), ie., if $\mathrm{Re}\lambda_i < 0$ for all eigenvalues of A.

Quadratic Lyapunov functions can also be used to investigate the stability of *linear differential inclusions* (LDIs), ie., the systems of type

$$\dot{\mathbf{x}} = A(t)\mathbf{x}, \;\; A(t) \in \Omega, \;\; \mathbf{x} \in \Re^n,$$

where $\Omega \subset \Re^{n \times n}$. This LDI can be interpreted as describing an uncertain time-varying linear system with the set Ω describing the uncertainty in the matrix $A(t)$. A special form of LDI is the *polytopic* LDI where $A(t) \in \mathbf{Co}\{A_1, ..., A_L\}$ where **Co** denotes the convex hull, ie.,

$$\mathbf{Co}\{A_1, ..., A_L\} = \{\sum_{i=1}^{p} \lambda_i X_i | X_i \in \{A_1, ..., A_L\}, \lambda_i \in \Re, p \geq 0\}.$$

Therefore $A(t)$ lies in the convex hull spanned by the matrices $A_1, ..., A_L$, which is equivalent to

$$\dot{\mathbf{x}} = \sum_{i=1}^{L} w_i(\mathbf{x}) A_i \mathbf{x}, \tag{4.8}$$

where $w_i(\mathbf{x})$ satisfies

$$0 \leq w_i(\mathbf{x}) \leq 1 \tag{4.9}$$

$$\sum_{i=1}^{L} w_i(\mathbf{x}) = 1. \tag{4.10}$$

The conditions (4.9) and (4.9) are known as the convexity conditions.

For polytopic LDI a sufficient stability condition can be formulated in the following way.

Theorem 4.2 *(Quadratic stability) The equilibrium* $\mathbf{x} = \mathbf{0}$ *of the polytopic LDI system (4.8) is asymptotically stable if there exists a common matrix P such that*

$$\begin{aligned} P &> 0 \\ A_i^T P + P A_i &< 0, \;\; i = 1, ...L. \end{aligned} \tag{4.11}$$

This is equivalent to saying that one must find a single Lyapunov function that holds for all the linear dynamic systems A_i involved.

The quadratic stability problem for polytopic LDIs has no analytic solution. However, it can be conveniently solved using a convex optimization technique (Boyd, Ghaoui, Feron and Balakrishnan, 1994).

In some cases it is also of interest to verify that no common solution P exists. This can be verified by solving the following dual problem (Boyd, Ghaoui, Feron and Balakrishnan, 1994; Johansson and Rantzer, 1998):

Theorem 4.3 *If there exist semi positive definite matrices R_i, not all zero, such that*

$$\sum_{i=1}^{L} A_i^T R_i + R_i A_i \geq 0,$$

then there is no such a positive definite matrix P to form a common quadratic Lyapunov function for all local models of (4.8).

Quadratic stability can be used to decide stability for continuous time linear TS systems, since it is easy to see that, if we choose $\mathbf{f}_i(\mathbf{x})$ to be linear model $A_i\mathbf{x}$, the continuous time state space TS model (2.20) is exactly a polytopic LDI.

4.3 Piecewise Lyapunov functions

When no globally quadratic Lyapunov function can be found, there are few methods for efficient construction of Lyapunov functions. For systems with local model structure, a natural and powerful extension of globally quadratic Lyapunov functions is to consider functions that are piecewise quadratic, since the dynamics given by local models are only valid within local areas. The search for a piecewise quadratic Lyapunov function can also be formulated as a convex optimization problem. As an illustrative example, we consider the state space TS models (2.19) and (2.20). Before starting, a few concepts need to be defined.

By the *support* Ω_i of $w_i(\mathbf{x}(t))$ we mean the set of states of (2.19) (or (2.20)) fulfilling:

$$\Omega_i = \{\mathbf{x} \in \Re^n | w_i(\mathbf{x}) > 0\}, \quad i \in I_L = \{1, ...L\}. \tag{4.12}$$

It is required that the state space of $\mathbf{x}$: $\Omega_1 \cup ... \cup \Omega_r = \Omega \subseteq \Re^n$, where $\mathbf{0} \in \Omega$. There is no loss of generality in assuming that the origin is the equilibrium point of (2.19)

(or (2.20)) since otherwise the equilibrium can be shifted to the origin by a simple translation of the state variables.

For brevity denote $w_i(\mathbf{x}(t))$ as $w_i(t)$, and define

$$Z_i = \{\mathbf{x} \in \Re^n | w_i(\mathbf{x}(t)) \geq w_j(\mathbf{x}(t)), j \in I_L\}, i \in I_L. \tag{4.13}$$

where $\{Z_1, ..., Z_r\}$ is a partition of the state space Ω which is determined by the dominance of the local subsystems (2.19) (or (2.20)).

The *switch region* Λ_{ij} is defined as

$$\Lambda_{ij} = \{\mathbf{x} \in \Re^n | \mathbf{x}(t^-) \in Z_i, \mathbf{x}(t) \in Z_j\}, \tag{4.14}$$

which is the set of states where the trajectory $\mathbf{x}(t)$ passes from Z_i to Z_j. Note that Λ_{ij} are given by hyper-surfaces.

Define

$$I_\Lambda = \{(i,j) | \Lambda_{ij} \neq \emptyset\}. \tag{4.15}$$

The concepts of Ω_i, Z_i and Λ_{ij} are illustrated in Figure 4.2 for the case of $\mathbf{x} = [x_1\ x_2]$ and triangular fuzzy membership functions. Note that while Ω_i $(i = 1,2,3,4)$ are overlaped areas, Z_i $(i = 1,2,3,4)$ forms a partition of the operating point space Ω. Also, the sets Λ_{12} and Λ_{43} are empty sets, ie., both Λ_{12} and Λ_{43} are not in I_Λ defined in (4.15) on this occasion.

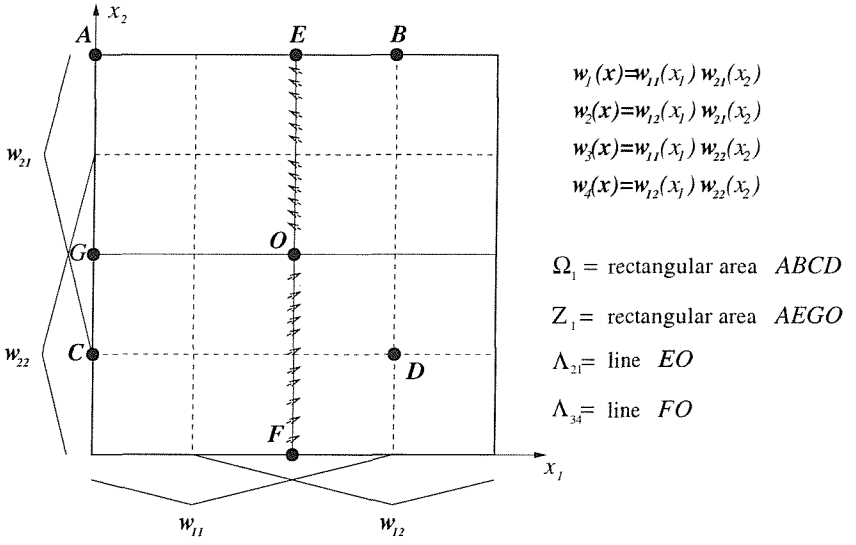

Figure 4.2: An example of Ω_i, Z_i and Λ_{ij}, the arrows show the vector field of the system.

We now have the following theorem about piecewise Lyapunov stability conditions (Pettersson and Lennartson, 1997a):

Theorem 4.4 *Assume that the state trajectory evolves according to (2.20). If there exist scalar functions* $V_i(\mathbf{x}) : Z_i \to \Re$, *each* $V_i(\mathbf{x})$ *differentiable in* $\mathbf{x}$, $i \in I_L$, *and constants* $\alpha > 0, \beta > 0$, *and* $\gamma > 0$ *such that*

- $\forall \mathbf{x} \in Z_i, \alpha\|\mathbf{x}\|^2 \leq V_i(\mathbf{x}) \leq \beta\|\mathbf{x}\|^2, i \in I_L$,
- $\forall \mathbf{x} \in Z_i, \dot{V}_i(\mathbf{x}) \leq -\gamma\|\mathbf{x}\|^2, i \in I_L$,
- $\forall \mathbf{x} \in \Lambda_{ij}, V_j(\mathbf{x}) \leq V_i(\mathbf{x}), (i,j) \in I_\Lambda$,

then the equilibrium point $\mathbf{0}$ *is exponentially stable in the sense of Lyapunov.*

If we define a global piecewise Lyapunov function by combining the local Lyapunov functions, the key of theorem 4.4 is to ensure that the global Lyapunov function does not increase at points in the switch regions. Another way to form a global Lyapunov function from local Lyapunov functions is to guarantee the continuity of the global Lyapunov function at points in the switch regions. Readers please refer to Johansson and Rantzer (1998) for further details.

4.4 Lyapunov stability of discrete time systems

This section states the corresponding stability results for discrete time systems.

Consider a discrete system described by

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t)), \tag{4.16}$$

where $\mathbf{x}(t) \in \Re^n, \mathbf{f}(\mathbf{x}(t))$ is an $n \times 1$ function vector with the property that

$$\mathbf{f}(\mathbf{0}) = \mathbf{0}.$$

We have the following corresponding stability definitions:

Definition 4.3 *The equilibrium point* $\mathbf{x} = \mathbf{0}$ *of (4.16) is*

- ***stable*** *if, for each* $R > 0$, *there exists* $r = r(R) > 0$ *such that*

$$\|\mathbf{x}_0\| < r \implies \|\mathbf{x}(t)\| < R, \ \forall t \geq 0,$$

- ***unstable*** *if not stable,*
- ***asymptotically stable*** *if it is stable and* r *can be chosen such that*

$$\|\mathbf{x}_0\| < r \implies \lim_{t\to\infty} \mathbf{x}(t) = 0,$$

- ***exponentially stable*** *if, for each* $R > 0$, *there exists* $r = r(R) > 0$ *and two numbers* $c_1 = c_1(R) > 0$ *and* $c_2 = c_2(R) > 0$ *such that*

$$\|\mathbf{x}_0\| < r \implies \|\mathbf{x}(t)\| < c_1 e^{-c_2 t}\|\mathbf{x}_0\|, \ \forall t \geq 0.$$

Definition 4.4 *The equilibrium point* $\mathbf{x} = \mathbf{0}$ *of (4.16) is*

- ***globally asymptotically stable*** *if it is asymptotically stable for any initial states.*
- ***globally exponentially stable*** *if it is exponentially stable for any initial states.*

Based on definitions 4.3 and 4.4, the following theorems can be obtained.

Theorem 4.5 *(Lyapunov's stability theorem—discrete time systems) (Kuo, 1980) Suppose* $\mathbf{x}$ *evolves according to (4.16). If there exists a scalar function* $V(\mathbf{x}(t))$ *continuous in* $\mathbf{x}(t)$ *such that*

1. $V(\mathbf{0}) = 0$,
2. $V(\mathbf{x}(t)) > 0$ *for* $\mathbf{x}(t) \neq \mathbf{0}$,
3. $V(\mathbf{x}(t))$ *approaches infinity as* $\|\mathbf{x}(t)\| \longrightarrow \infty$,
4. $\triangle V(\mathbf{x}(t)) = V(\mathbf{x}(t+1)) - V(\mathbf{x}(t)) < 0$ *for* $\mathbf{x}(t) \neq \mathbf{0}$.

Then the equilibrium state $\mathbf{x} = \mathbf{0}$ *is asymptotically stable in the large and* $V(\mathbf{x}(t))$ *is a Lyapunov function.*

Choosing the local models in (2.19) to be linear models, we obtain the following *discrete time state space linear TS model*:

$$\mathbf{x}(t+1) = \sum_{i=1}^{L} w_i(\mathbf{x}(t))A_i\mathbf{x}(t). \tag{4.17}$$

The quadratic Lyapunov stability theorem for (4.17) can be state as follows (Tanaka and Sugeno, 1992).

Theorem 4.6 *The equilibrium of a fuzzy system (4.17) is globally asymptotically stable if there exists a common positive definite matrix* P *for all the subsystems such that*

$$A_i^T P A_i - P < 0 \ \ for \ \ i \in I_L = \{1, \ldots, L\}. \tag{4.18}$$

The conditions of a piecewise Lyapunov function for system (2.19) can be state as (Feng and Harris, 1999b)

Theorem 4.7 *Assume that the state trajectory evolves according to (2.19). If there exist scalar functions* $V_i(\mathbf{x}) : Z_i \to \Re$, *each* $V_i(\mathbf{x})$ *differentiable in* $\mathbf{x}$, $i \in I_L$, *and constants* $\alpha > 0, \beta > 0$, *and* $\gamma > 0$ *such that*

- $\forall \mathbf{x} \in Z_i, \alpha\|\mathbf{x}\|^2 \leq V_i(\mathbf{x}) \leq \beta\|\mathbf{x}\|^2, i \in I_L$,
- $\forall \mathbf{x} \in Z_i, V_i(\mathbf{x}(t+1)) - V_i(\mathbf{x}(t)) \leq -\gamma\|\mathbf{x}(t)\|^2, i \in I_L$,
- $\forall \mathbf{x}(\in \Lambda_{ij}, V_j(\mathbf{x}(t)) \leq V_i(\mathbf{x}(t^-)), (i,j) \in I_\Lambda$,

then the equilibrium point $\mathbf{0}$ *is exponentially stable in the sense of Lyapunov.*

Proof: Define a piecewise differentiable function $V(\mathbf{x})$ as

$$V(\mathbf{x}) = \begin{cases} V_i(\mathbf{x}) & \text{if } \mathbf{x} \in Z_i \text{ but } \mathbf{x} \text{ not in } \Lambda_{qr} \text{ for any } (q,r) \in I_\Lambda, \\ V_q(\mathbf{x}) & \text{if } \mathbf{x} \in \Lambda_{qr}, \text{ where } (q,r) \in I_\Lambda. \end{cases} \tag{4.19}$$

Suppose $\mathbf{x}(0) = \mathbf{x}_0$. Without lose of generality, suppose $\beta > \gamma$, since if not, we can choose a bigger $\beta_1 > \gamma$ and replace the β in the first condition with β_1. Define $c_1 = (\frac{\beta}{\alpha})^{\frac{1}{2}}$, $c_2 = -\frac{1}{2}\ln(1 - \frac{\gamma}{\beta})$. then $c_2 > 0$. Firstly let us prove that

$$V(\mathbf{x}(t)) \leq V(\mathbf{x}_0)e^{-2c_2 t}, \quad \forall t \geq 0 \tag{4.20}$$

is true by induction. Assume that the trajectory is in region Z_p in the time interval $t \in [0\ t_1)$. Then according to the definition (4.19), $V(\mathbf{x}(t)) = V_p(\mathbf{x}(t))$, $t \in [0\ t_1)$. Using the first and second condition of the theorem we have

$$V(\mathbf{x}(t)) - V(\mathbf{x}(t-1)) \leq -\gamma\|\mathbf{x}(t-1)\|^2 \leq -\frac{\gamma}{\beta}V(\mathbf{x}(t-1)), \quad t \in [0\ t_1). \tag{4.21}$$

Consequently

$$V(\mathbf{x}(t)) \leq \left(1 - \frac{\gamma}{\beta}\right)^t V(\mathbf{x}_0) = V(\mathbf{x}_0)e^{-2c_2 t}, \quad t \in [0\ t_1). \tag{4.22}$$

If t_1 is infinite, meaning that the trajectory never leaves the region Z_p, then (4.20) is true. Otherwise, assume that the trajectory passes through different regions and stays in Z_q for $t \in [t_k\ t_{k+1})$. Assume that

$$V(\mathbf{x}(t)) \leq V(\mathbf{x}_0)e^{-2c_2 t}, \ \forall t \in [t_k\ t_{k+1}). \tag{4.23}$$

Note that according to the definition (4.19) $V(\mathbf{x}(t)) = V_q(\mathbf{x}(t))$ for $t \in [t_k\ t_{k+1})$. Suppose the trajectory reaches Λ_{qr} at time t_{k+1} and stays in the region Z_r for $t \in [t_{k+1}\ t_{k+2})$, where t_{k+2} maybe infinite. Using the first and second condition of the theorem,

$$V(\mathbf{x}(t)) - V(\mathbf{x}(t-1)) \leq -\gamma\|\mathbf{x}(t-1)\|^2 \leq -\frac{\gamma}{\beta}V(\mathbf{x}(t-1)), \quad t \in [t_{k+1}\ t_{k+2}).$$

Therefore

$$V(\mathbf{x}(t)) \leq V(\mathbf{x}(t_{k+1}))e^{-2c_2(t-t_{k+1})}, \ \forall t \in [t_{k+1}\ t_{k+2}). \tag{4.24}$$

Note that in (4.24) $V(\mathbf{x}(t_{k+1})) = V_r(\mathbf{x}(t_{k+1})) \leq V_q(\mathbf{x}(t_{k+1}))$ according to the third condition of the theorem. Therefore from (4.24) and (4.23),

$$\begin{aligned} V(\mathbf{x}(t)) &\leq V(\mathbf{x}(t_{k+1}))e^{-2c_2(t-t_{k+1})} \\ &\leq V_q(\mathbf{x}(t_{k+1}^-))e^{-2c_2(t-t_{k+1})} \\ &\leq V(\mathbf{x}_0)e^{-2c_2t_{k+1}}e^{-2c_2(t-t_{k+1})} \\ &= V(\mathbf{x}_0)e^{-2c_2t}, \quad \forall t \in [t_{k+1}\ t_{k+2}). \end{aligned} \tag{4.25}$$

Combining (4.23) and (4.25) it follows that

$$V(\mathbf{x}(t)) \leq V(\mathbf{x}_0)e^{-2c_2t}, \ \forall t \in [t_k\ t_{k+2}). \tag{4.26}$$

It can be concluded from the principle of induction that (4.20) is true. According to the first condition of the theorem and (4.20),

$$\begin{aligned} \|\mathbf{x}(t)\| &\leq \left(\frac{1}{\alpha}V(\mathbf{x}(t))\right)^{\frac{1}{2}} \\ &\leq \left(\frac{1}{\alpha}V(\mathbf{x}_0)e^{-2c_2t}\right)^{\frac{1}{2}} \\ &\leq \left(\frac{1}{\alpha}\beta\|\mathbf{x}_0\|^2e^{-2c_2t}\right)^{\frac{1}{2}} \\ &= c_1e^{-c_2t}\|\mathbf{x}_0\| \quad \forall t > 0. \end{aligned}$$

According to the definitions 4.3 and 4.4 of exponentially stability, the theorem has been established.

4.5 Linear matrix inequalities

The first *linear matrix inequality* (LMI) used to analyse stability of a dynamical system was the Lyapunov inequality (4.7) which was investigated by Lyapunov himself.

The important role of LMIs in system and control theory was already recognized in the early 1960's, especially by Yakubovich (Yakubovich, 1962; Yakubovich, 1964; Yakubovich, 1967). In 1984, Karmarkar (1984) introduced a new linear programming algorithm that solves linear programs in polynomial-time, which is very efficient in practice. Karmarkar's work spurred an enormous amount of work in the area of interior-point methods for linear programming. Then in 1988, Nesterov (Nesterov and Nemirovsky, 1988; Nesterov and Nemirovsky, 1994) developed interior-point methods that apply directly to convex problems involving LMIs. Although there remains much to be done in this area, several interior-point algorithms for LMI problems have been implemented and tested on specific families of LMIs that arise in system and control theory, and found to be extremely efficient. Nowadays efficient computerized methods to solve such problems are available in commercially supported software such as MATLAB (Gahinet, Nemirovski, Laub and Chilali, 1995). In the following we introduce some basic concepts of LMIs concerning the topics of this thesis.

A general LMI has the form

$$F(\mathbf{x}) = F_0 + \sum_{i=1}^{m} x_i F_i > 0, \tag{4.27}$$

where $\mathbf{x} \in \Re^m$ is the variable and the symmetric matrices $F_i = F_i^T \in \Re^{n\times n}, i = 0, 1, ..., m$, are given. The inequality symbol in (4.27) means that $F(\mathbf{x})$ is positive definite. Of course, the LMI (4.27) is equivalent to a set of n polynomial inequalities in x_i, ie., the leading principal minors of $F(\mathbf{x})$ must be positive.

We will also encounter *nonstrict* LMIs, which have the form

$$F(\mathbf{x}) \geq 0, \tag{4.28}$$

The LMI (4.27) is a convex constraint on $\mathbf{x}$, ie., the set $\{\mathbf{x}|F(\mathbf{x}) > 0\}$ is convex. Although the LMI (4.27) may seem to have a specialized form, it can represent a wide variety of convex constraints on $\mathbf{x}$. In particular, linear inequalities, (convex) quadratic inequalities, matrix norm inequalities, and constraints that arise in control theory, such as Lyapunov and convex quadratic matrix inequalities, can all be cast in the form of an LMI.

Multiple LMIs $F_1(\mathbf{x}) > 0, ..., F_p(\mathbf{x}) > 0$ can be expressed as a single LMI $\text{diag}(F_1(\mathbf{x}), ..., F_p(\mathbf{x})) > 0$ where $\text{diag}(\cdot)$ means diagonal matrix. Therefore we will make no distinction between a set of LMIs and a single LMI, ie., "the LMIs $F_1(\mathbf{x}) > 0, ..., F_p(\mathbf{x}) > 0$" will mean "the LMI $\text{diag}(F_1(\mathbf{x}), ..., F_p(\mathbf{x})) > 0$". We also note that when the matrix F_i are diagonal, the LMI $F(\mathbf{x}) > 0$ is just a set of linear inequalities.

4.5.1 The $\mathcal{S}$-procedure

The $\mathcal{S}$-procedure (Aizerman and Gantmacher, 1964; Aiserman and Gantmacher, 1965) is a technique to replace a condition on a function with constraints by a condition without constraints. When dealing with local Lyapunov stability problems, we often encounter the constraint that some quadratic function (or quadratic form) be negative whenever some other quadratic functions (or quadratic forms) are all negative, for example, if some conditions must be satisfied in some special regions such as the conditions in theorem 4.4 and theorem 4.7. In some cases, this constraint can be expressed as an LMI in the data defining the quadratic functions or forms; in other cases, we can form an LMI that is a conservative but often useful approximation of the constraint. In the following we describe two common versions of the $\mathcal{S}$-procedure.

4.5.1.1 The $\mathcal{S}$-procedure for quadratic functions and nonstrict inequalities

Let $F_0,...,F_p$ be quadratic functions of a variable $\mathbf{x} \in \Re^n$. ie.,

$$F_i(\mathbf{x}) = \mathbf{x}^T M_i \mathbf{x} + 2\mathbf{u}_i^T \mathbf{x} + v_i, \quad i = 0,...,p,$$

where $M_i = M_i^T$. Consider the following condition on $F_0,...,F_p$:

$$F_0(\mathbf{x}) \geq 0 \quad \text{for } \{\mathbf{x} \in \Re^n | F_i(\mathbf{x}) \geq 0 \; i = 1,...,p\}. \tag{4.29}$$

Obviously if there exist $\tau_1 \geq 0,...,\tau_p \geq 0$ such that

$$\forall \mathbf{x} \in \Re^n, F_0(\mathbf{x}) \geq \sum_{i=1}^{p} \tau_i F_i(\mathbf{x}), \tag{4.30}$$

then (4.29) holds. When $p = 1$, the converse holds provided that there is some $\mathbf{x}_0$ such that $F_1(\mathbf{x}_0) > 0$.

If the functions F_i are affine, ie.,

$$F_i(\mathbf{x}) = \mathbf{u}_i^T \mathbf{x} + v_i,$$

then according to *Farkas lemma* (Boyd, Ghaoui, Feron and Balakrishnan, 1994), (4.29) and (4.30) are equivalent. That is to say,

$$\mathbf{u}_0^T \mathbf{x} + v_0 \geq 0 \quad \text{for } \{\mathbf{x} \in \Re^n | \mathbf{u}_i^T \mathbf{x} + v_i \geq 0 \; i = 1,...,p\}$$

if and only if there exist $\tau_1 \geq 0,...,\tau_p \geq 0$ such that

$$\forall \mathbf{x} \in \Re^n, \mathbf{u}_0^T \mathbf{x} + v_0 \geq \sum_{i=1}^{p} \tau_i (\mathbf{u}_i^T \mathbf{x} + v_i).$$

Note that (4.30) can be written as an LMI:

$$\begin{bmatrix} M_0 & u_0 \\ u_0^T & v_0 \end{bmatrix} - \sum_{i=1}^{p} \tau_i \begin{bmatrix} M_i & u_i \\ u_i^T & v_i \end{bmatrix} > 0.$$

4.5.1.2 The $\mathcal{S}$-procedure for quadratic forms and strict inequalities

Let $M_0, ..., M_p \in \Re^{n \times n}$ be symmetric matrices. We consider the following condition on $T_0, ..., T_p$:

$$\mathbf{x}^T M_0 \mathbf{x} > 0 \text{ for all } \mathbf{x} \neq 0 \text{ such that } \mathbf{x}^T M_i \mathbf{x} \geq 0, \quad i = 1, ..., p. \tag{4.31}$$

It is obvious that if there exist $\tau_1 \geq 0, ..., \tau_p \geq 0$ such that

$$M_0 - \sum_{i=1}^{p} \tau_i M_i > 0 \tag{4.32}$$

then (4.31) holds. Similarly as the last section, when $p = 1$, the converse holds provided that there exists some $\mathbf{x}_0$ such that $\mathbf{x}_0^T M_1 \mathbf{x}_0 > 0$. Note that (4.32) is an LMI in the variables M_0 and $\tau_1, ...\tau_p$.

The first version of the $\mathcal{S}$-procedure deals with nonstrict inequalities and quadratic functions that may include constant and linear terms. The second version deals with strict inequalities and quadratic forms only, ie., quadratic functions without constant or linear terms.

4.5.2 Schur complements

Schur complements are used to convert certain *nonlinear matrix inequalities* to LMIs. We state the basic result of Schur complements as the following lemma (Boyd, Ghaoui, Feron and Balakrishnan, 1994):

Lemma 4.1 *(Schur Complements Lemma) The LMI*

$$\begin{bmatrix} Q(\mathbf{x}) & S(\mathbf{x}) \\ S(\mathbf{x})^T & R(\mathbf{x}) \end{bmatrix} > 0, \tag{4.33}$$

where $Q(\mathbf{x}) = Q(\mathbf{x})^T$, $R(\mathbf{x}) = R(\mathbf{x})^T$, and $S(\mathbf{x})$ depend affinely on $\mathbf{x}$, is equivalent to

$$R(\mathbf{x}) > 0, \quad Q(\mathbf{x}) - S(\mathbf{x})R(\mathbf{x})^{-1}S(\mathbf{x})^T > 0. \tag{4.34}$$

In other words, the set of nonlinear inequalities (4.34) can be represented as the LMI (4.33).

The preceding Schur complement lemma can be generalized to nonstrict inequalities (Boyd, Ghaoui, Feron and Balakrishnan, 1994). Suppose Q and R are symmetric. The condition

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \geq 0, \tag{4.35}$$

is equivalent to

$$R \geq 0, \quad Q - SR^{\dagger}S^T \geq 0, \quad S(I - RR^{\dagger}) = 0, \tag{4.36}$$

where $R^{\dagger}$ denotes the Moore-Penrose inverse of R.

4.6 Conclusions

In this chapter, the existing Lyapunov stability results of both continuous and discrete time systems concerning our work have been presented. Quadratic Lyapunov functions for linear systems and piecewise Lyapunov functions for local systems are discussed in detail. A strict proof of piecewise Lyapunov stability conditions for discrete time system has been given. Aspects of LMIs concerning Lyapunov stability, such as the $\mathcal{S}$-procedure and Schur complements are also discussed in this chapter. Next, in chapter 5, we will use all of the results presented in this chapter to derive new efficient Lyapunov stability conditions for local system model structure.

Chapter 5

Piecewise Lyapunov Stability Conditions of Fuzzy Systems

Stability plays a fundamental role in system and control theory. For fuzzy and neurofuzzy systems, stability analysis has been difficult because these systems are both nonlinear and represent linguistic/symbolic knowledge in terms of rules with variables that encapsulate vague or imprecise notions. Recently, some stability results for fuzzy systems have been reported: An investigation of input/output data signal based stability of a given direct static multiple-input single-output neuro-fuzzy controller operating under feedback control has been developed in French and Rogers (1998). For Lyapunov stability, one of the more recent approaches (Takagi and Sugeno, 1985) to determining stability of fuzzy/neurofuzzy systems is to decompose the global process into a series of local models/subsystems represented as TS models. Theorems 4.2 and 4.6 introduce sufficient conditions for the asymptotic stability of fuzzy systems in the sense of Lyapunov through the existence of a common Lyapunov function for all the subsystems (Tanaka and Sugeno, 1992); Whereas a general method for the computation of piecewise quadratic Lyapunov functions for hybrid systems which include Takagi-Sugeno fuzzy systems as a special case has also been derived (Johansson and Rantzer, 1998). Compared with the global results of theorem 4.6, the 'local' results of Johansson and Rantzer (1998) is significant, in that it searches for different quadratic Lyapunov function in different operating regions in state space and so significantly relaxes the stability conditions for a global fuzzy system. However, there are many situations where continuous Lyapunov functions are too restrictive; So a construction method (Pettersson and Lennartson, 1997b) for generating stability conditions of hybrid systems using discontinuous Lyapunov functions has been derived. Both local and hybrid methods of Johansson and Rantzer (1998) and Pettersson and

Lennartson (1997b) are attractive since the search for local Lyapunov functions can be reformulated as a set of LMIs.

A drawback that limits the practical use of the methods presented in Johansson and Rantzer (1998) and Pettersson and Lennartson (1997b) is that it may be required to solve a large number of LMIs in the interpolation regions between the system submodels. In addition to the high number of LMIs, the computation complexity and cost also increases dramatically as the input dimensionality increases. This means that the number of model parameters involved in the optimization process becomes prohibitively large for large dimensional systems. Also, for fuzzy membership functions with global support the stability conditions reduce to the case of global quadratic stability, hence the preference for basis functions such as B-splines with compact support. The majority of current methods for stability analysis of fuzzy systems ignore the membership function characteristics, which contain important structural information about the system contained in the rule premises. Many of the methods simply restrict membership functions to triangular functions (with compact support) for ease of analysis, leading to piecewise linear control surfaces—albeit with discontinuities.

In this chapter, we present a new method for the stability analysis of fuzzy systems that incorporates the input membership function characteristics. We show that, under certain conditions placed on the input membership functions, we need only search for one local Lyapunov function even in the intermodel interpolation region. This both relaxes the stability conditions and reduces the computation load in solving the resultant reduced number of LMIs.

5.1 System description

The Takagi-Sugeno fuzzy model is considered in this chapter since it has established modelling, stability and control conditions. Both the discrete time state space model (2.19) and continuous time state space model (2.20) are considered. That is to say, the model can be both continuous time:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{L} w_i(t)[A_i\mathbf{x}(t) + B_i], \tag{5.1}$$

or discrete time

$$\mathbf{x}(t+1) = \sum_{i=1}^{L} w_i(t)[A_i\mathbf{x}(t) + B_i], \tag{5.2}$$

where $w_i : \Re^n \to \Re^+, i \in I_L = \{1,...,L\}$ are normalized membership functions with $w_i(t) \geq 0$ for $i \in I_L$ and

$$\sum_{i=1}^{L} w_i(t) = 1, \forall t.$$

We will use the definitions of Ω_i, Z_i, Λ_{ij} and I_Λ which are defined in (4.12)—(4.15). Recall that it is required that the state space of $\mathbf{x}$: $\Omega_1 \cup ... \cup \Omega_r = \Omega \subseteq \Re^n$, where $\mathbf{0} \in \Omega$ and there is no loss of generality in assuming that the origin is the equilibrium point of (5.1) and (5.2). Because A_i is a matrix with constant entries, $A_i\mathbf{x}(t) + B_i$ is an affine system defined on Ω_i. We note that origin being the equilibrium implies that,

$$\sum_{i=1}^{L} w_i(\mathbf{0})B_i = \mathbf{0}.$$

For brevity denote $w_i(\mathbf{x}(t))$ as $w_i(t)$, and introduce

$$\bar{A}_i = \begin{bmatrix} A_i & B_i \\ 0 & 0 \end{bmatrix}, \quad i \in I_L, \qquad \bar{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \in \Re^{n+1}.$$

Within each subregion Z_i, the fuzzy system (5.1) can be rewritten as:

$$\begin{aligned} \dot{\bar{\mathbf{x}}}(t) &= [\bar{A}_i + \sum_{j=1}^{L} w_j(t)(\bar{A}_j - \bar{A}_i)]\bar{\mathbf{x}}(t) \\ &= \bar{A}_i\bar{\mathbf{x}}(t) + \sum_{j=1}^{L} w_j(t)\Delta\bar{A}_{ij}\bar{\mathbf{x}}(t) \\ &= \bar{A}_i\bar{\mathbf{x}}(t) + \mathbf{h}_i(t,\mathbf{x}(t)), \end{aligned} \tag{5.3}$$

where

$$\begin{aligned} \Delta\bar{A}_{ij} &= \bar{A}_j - \bar{A}_i = \begin{bmatrix} A_j - A_iB_j - B_i \\ 0 \quad 0 \end{bmatrix}, \\ \mathbf{h}_i(t,\mathbf{x}(t)) &= \sum_{j=1}^{L} w_j(t)\Delta\bar{A}_{ij}\bar{\mathbf{x}}(t). \end{aligned}$$

Accordingly, within Z_i, the fuzzy system (5.2) can be rewritten as:

$$\bar{\mathbf{x}}(t+1) = \bar{A}_i\bar{\mathbf{x}}(t) + \mathbf{h}_i(t,\mathbf{x}(t)). \tag{5.4}$$

Note that $\mathbf{h}_i(t,\mathbf{x}(t))$ can be viewed as a kind of modelling error of the dominate local model in the subregion Z_i.

5.2 Stability conditions

Applying theorem 4.4 or theorem 4.7 to perform a stability analysis of the system (5.1) and (5.2), we are going to find a type of discontinuous Lyapunov function. This means that we have to construct the different Lyapunov functions $V_i, i \in I_L$, satisfying the specified conditions in the subspace $Z_i, i \in I_L$ and $\Lambda_{ij}, (i,j) \in I_\Lambda$. This can be achieved by re-stating the stability conditions as LMIs. We express $V_i, i \in I_L$ with quadratic local Lyapunov functions according to

$$V_i(\mathbf{x}) = \mathbf{x}^T P_i \mathbf{x} + 2\mathbf{p}_i^T \mathbf{x} + q_i = \bar{\mathbf{x}}^T \bar{P}_i \bar{\mathbf{x}}, \tag{5.5}$$

where

$$\bar{P}_i = \begin{bmatrix} P_i & \mathbf{p}_i \\ \mathbf{p}_i^T & q_i \end{bmatrix},$$

and $P_i^T = P_i \in \Re^{n\times n}$, $\mathbf{p}_i \in \Re^{n\times 1}$ and $q_i \in \Re$.

From (5.3) and (5.5) we get

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \dot{\bar{\mathbf{x}}}^T \bar{P}_i \bar{\mathbf{x}} + \bar{\mathbf{x}}^T \bar{P}_i \dot{\bar{\mathbf{x}}} \\ &= [\bar{\mathbf{x}}^T \bar{A}_i + \mathbf{h}_i^T(t,\mathbf{x})]\bar{P}_i \bar{\mathbf{x}} + \bar{\mathbf{x}}^T \bar{P}_i [\bar{A}_i \bar{\mathbf{x}} + \mathbf{h}_i(t,\mathbf{x})] \\ &= \bar{\mathbf{x}}^T (\bar{A}_i^T \bar{P}_i + \bar{P}_i \bar{A}_i)\bar{\mathbf{x}} + \mathbf{h}_i^T(t,\mathbf{x})\bar{P}_i \bar{\mathbf{x}} + \bar{\mathbf{x}}^T \bar{P}_i \mathbf{h}_i(t,\mathbf{x}) \\ &\leq \bar{\mathbf{x}}^T (\bar{A}_i^T \bar{P}_i + \bar{P}_i \bar{A}_i)\bar{\mathbf{x}} + 2\bar{\mathbf{x}}^T \bar{P}_i \sum_{j=1}^{L} \xi_{ij} \bar{\mathbf{x}} \\ &= \bar{\mathbf{x}}^T (\bar{A}_i^T \bar{P}_i + \bar{P}_i \bar{A}_i + 2\bar{P}_i \sum_{j=1}^{L} \xi_{ij})\bar{\mathbf{x}}, \end{aligned} \tag{5.6}$$

where

$$\xi_{ij} = \|\Delta \bar{A}_{ij}\| sup_{\mathbf{x}\in Z_i} w_j(\mathbf{x}(t)), \tag{5.7}$$

and $\|\cdot\|$ is the square norm of a matrix.

Analogously, for the discrete-time model (5.2), from (5.4) and (5.5) we get

$$V(\mathbf{x}(t+1)) - V(\mathbf{x}(t)) = \bar{\mathbf{x}}^T(t)(\bar{A}_i^T \bar{P}_i \bar{A}_i - \bar{P}_i + 2\bar{A}_i^T \bar{P}_i \sum_{j=1}^{L} \xi_{ij} + \bar{P}_i (\sum_{j=1}^{L} \xi_{ij})^2)\bar{\mathbf{x}}(t), \tag{5.8}$$

where again ξ_{ij} is expressed in (5.7).

The following theorem for continuous-time model (5.1) is obtained from equation (5.6) and theorem 4.4 directly:

Theorem 5.1 *Assume that the state trajectory evolves according to (5.1). If there exist* $\bar{P}_i(\mathbf{x}) = \bar{P}_i^T(\mathbf{x}), i \in I_L$, *and constants* $\alpha > 0, \beta > 0$, *and* $\gamma > 0$ *such that*

(i) $\forall \mathbf{x} \in Z_i, \alpha\|\bar{\mathbf{x}}\|^2 \leq \bar{\mathbf{x}}^T \bar{P}_i \bar{\mathbf{x}} \leq \beta\|\bar{\mathbf{x}}\|^2, i \in I_L$,

(ii) $\forall \mathbf{x} \in Z_i, \bar{\mathbf{x}}^T(\bar{A}_i^T \bar{P}_i + \bar{P}_i \bar{A}_i)\bar{\mathbf{x}} + 2\bar{\mathbf{x}}^T \bar{P}_i \sum_{j=1}^L \xi_{ij}\bar{\mathbf{x}} \leq -\gamma\|\bar{\mathbf{x}}\|^2, i \in I_L$,

(iii) $\forall \mathbf{x} \in \Lambda_{ij}, \bar{\mathbf{x}}^T \bar{P}_j \bar{\mathbf{x}} \leq \bar{\mathbf{x}}^T \bar{P}_i \bar{\mathbf{x}}, (i,j) \in I_\Lambda$,

then the equilibrium point $\mathbf{0}$ *is exponentially stable in the sense of Lyapunov.*

For the discrete-time model (5.2), from (5.8) and theorem 4.7 we have:

Theorem 5.2 *Assume that the state trajectory evolves according to (5.2). If there exist* $\bar{P}_i(\mathbf{x}) = \bar{P}_i^T(\mathbf{x}), i \in I_L$, *and constants* $\alpha > 0, \beta > 0$, *and* $\gamma > 0$ *such that*

(i) $\forall \mathbf{x} \in Z_i, \alpha\|\bar{\mathbf{x}}\|^2 \leq \bar{\mathbf{x}}^T \bar{P}_i \bar{\mathbf{x}} \leq \beta\|\bar{\mathbf{x}}\|^2, i \in I_L$,

(ii) $\forall \mathbf{x} \in Z_i, \bar{\mathbf{x}}^T(\bar{A}_i^T \bar{P}_i \bar{A}_i - \bar{P}_i + 2\bar{A}_i^T \bar{P}_i \sum_{j=1}^L \xi_{ij} + \bar{P}_i(\sum_{j=1}^L \xi_{ij})^2)\bar{\mathbf{x}} \leq -\gamma\|\bar{\mathbf{x}}\|^2, i \in I_L$,

(iii) $\forall \mathbf{x} \in \Lambda_{ij}, \bar{\mathbf{x}}^T \bar{P}_j \bar{\mathbf{x}} \leq \bar{\mathbf{x}}^T \bar{P}_i \bar{\mathbf{x}}, (i,j) \in I_\Lambda$,

then the equilibrium point $\mathbf{0}$ *is exponentially stable in the sense of Lyapunov.*

5.3 LMI formulation

As discussed in chapter 4, the main obstacle to a direct application of the Lyapunov techniques is the nontrivial step of finding the appropriate Lyapunov function for a nonlinear system. Nevertheless, the conditions in theorem 5.1 and theorem 5.2 can be reformulated in the form of LMIs described in chapter 4 to enable us search for the solutions by optimal techniques. Before doing so, attention must be paid that the conditions only have to be satisfied in specified regions Z_i, whereas an LMI is a global condition. If the conditions are converted globally when reformulated as LMIs, then the results would be much more conservative. To solve the problem, we can utilize the S-procedure described in chapter 4 to express the different conditions in theorem 5.1 and theorem 5.2 as LMIs.

The regions $Z_i, i \in I_L$ and $\Lambda_{ij}, (i,j) \in I_\Lambda$ are hyperplanes in the space $\Re^n$. Therefore it can be assumed that

$$Z_i = \{\mathbf{x} \in \Re^n | Q_{ik}^T \mathbf{x} + d_{ik} \geq 0 \ k \in I_i\}, \tag{5.9}$$

$$\Lambda_{ij} = \{\mathbf{x} \in \Re^n | R_{ij}^T \mathbf{x} + c_{ij} = 0\}, \tag{5.10}$$

where Q_{ik} and R_{ij} are $n \times n$ matrices, d_{ik} and c_{ij} are scalar numbers. Define

$$\bar{Q}_{ik} = \begin{bmatrix} 0 & \frac{1}{2}Q_{ik} \\ \frac{1}{2}Q_{ik}^T & d_{ik} \end{bmatrix}, \qquad \bar{R}_{ij} = \begin{bmatrix} 0 & \frac{1}{2}R_{ij} \\ \frac{1}{2}R_{ij}^T & c_{ij} \end{bmatrix}.$$

Using the $\mathcal{S}$-procedure, the following theorem can be derived from theorem 5.1:

Theorem 5.3 *Assume that the state trajectory evolves according to (5.1). If there exist* $\bar{P}_i \in \Re^{(n+1)\times(n+1)}, i \in I_L$, *and constants* $\alpha > 0, \beta > 0, \gamma > 0$, *and* $\mu_{ik} \geq 0, \nu_{ik} \geq 0$, *and* η_{ij} *such that*

(i) $\alpha I + \sum_k \mu_{ik}\bar{Q}_{ik} \leq \bar{P}_i \leq \beta I - \sum_k \mu_{ik}\bar{Q}_{ik}, \ i \in I_L,$

(ii) $\bar{A}_i^T\bar{P}_i + \bar{P}_i\bar{A}_i + 2\bar{P}_i\sum_{j=1}^{L}\xi_{ij} + \sum_k \nu_{ik}\bar{Q}_{ik} \leq -\gamma I, \ i \in I_L,$

(iii) $\bar{P}_j \leq \bar{P}_i - \eta_{ij}\bar{R}_{ij}, \ (i,j) \in I_\Lambda,$

where I is the identity matrix, then the equilibrium point $\mathbf{0}$ *is exponentially stable in the sense of Lyapunov.*

Proof: $\forall i \in I_L$, firstly let's consider the inequality

$$\forall \mathbf{x} \in Z_i, \alpha\|\bar{\mathbf{x}}\|^2 \leq \bar{\mathbf{x}}^T\bar{P}_i\bar{\mathbf{x}}. \tag{5.11}$$

of theorem 5.1. According to the definitions of $\bar{Q}_{ik}$ and $\bar{R}_{ij}$, (5.9) can be rewritten as

$$Z_i = \{\bar{\mathbf{x}} \in \Re^{n+1} | \bar{\mathbf{x}}^T\bar{Q}_{ik}\bar{\mathbf{x}} \geq 0, \ k \in I_i\}. \tag{5.12}$$

Using (5.12), the condition (5.11) is equivalent to

$$\bar{\mathbf{x}}^T\bar{P}_i\bar{\mathbf{x}} - \alpha\|\bar{\mathbf{x}}\|^2 \geq 0 \ \text{ for } \ \{\bar{\mathbf{x}} \in \Re^{n+1} | \bar{\mathbf{x}}^T\bar{Q}_{ik}\bar{\mathbf{x}} \geq 0, \ k \in I_i\}. \tag{5.13}$$

Noting that $\bar{\mathbf{x}}\bar{Q}_{ik}\bar{\mathbf{x}}$ is affine in $\bar{\mathbf{x}}$, according to *Farkas lemma* stated in section 4.5.1.1, (5.13) is equivalent to the following condition, ie., there exist $\mu_{ik} \geq 0, \ k \in I_i$ such that

$$\forall \bar{\mathbf{x}} \in \Re^{n+1}, \ \bar{\mathbf{x}}^T\bar{P}_i\bar{\mathbf{x}} - \alpha\|\bar{\mathbf{x}}\|^2 \geq \sum_k \mu_{ik}\bar{\mathbf{x}}^T\bar{Q}_{ik}\bar{\mathbf{x}},$$

which can be rewritten as

$$\forall \bar{\mathbf{x}} \in \Re^{n+1}, \ \alpha\|\bar{\mathbf{x}}\|^2 + \sum_k \mu_{ik}\bar{\mathbf{x}}^T\bar{Q}_{ik}\bar{\mathbf{x}} \leq \bar{\mathbf{x}}^T\bar{P}_i\bar{\mathbf{x}}.$$

Since the above condition holds in the whole space, it can be rewritten as an LMI as following

$$\alpha I + \sum_k \mu_{ik}\bar{Q}_{ik} \leq \bar{P}_i,$$

which is the left part of the first condition in the theorem. All other parts of the conditions can be obtained similarly. The theorem established.

Similarly, from theorem 5.2 the following theorem can be established:

Theorem 5.4 *Assume that the state trajectory evolves according to (5.1). If there exist* $\bar{P}_i \in \Re^{(n+1)\times(n+1)}, i \in I_L$, *and constants* $\alpha > 0, \beta > 0, \gamma > 0$, *and* $\mu_{ik} \geq 0, \nu_{ik} \geq 0$, *and* η_{ij} *such that*

(i) $\alpha I + \sum_k \mu_{ik}\bar{Q}_{ik} \leq \bar{P}_i \leq \beta I - \sum_k \mu_{ik}\bar{Q}_{ik}, \quad i \in I_L,$

(ii) $\bar{A}_i^T \bar{P}_i \bar{A}_i - \bar{P}_i + 2\bar{A}_i^T \bar{P}_i \sum_{j=1}^L \xi_{ij} + \bar{P}_i (\sum_{j=1}^L \xi_{ij})^2 \leq -\gamma I, \quad i \in I_L,$

(iii) $\bar{P}_j \leq \bar{P}_i - \sum_k \eta_{ij}\bar{R}_{ij}, \quad (i,j) \in I_\Lambda,$

where I is the identity matrix, then the equilibrium point $\mathbf{0}$ *is exponentially stable in the sense of Lyapunov.*

Theorem 5.3 and theorem 5.4 give the stability conditions of system (5.1) and (5.2) respectively re-formulated as LMIs. Hence, it is computationally possible to find the unknown variables $\bar{P}_i$ in the theorems by efficient convex optimization algorithms, which is illustrated by the following examples.

5.4 Simulation examples

5.4.1 Example 1

This example is adopted from Johansson and Rantzer (1998) to compare the efficiencies of the proposed approach and the methods given in Johansson and Rantzer (1998) and Pettersson and Lennartson (1997b).

Assuming that some local modelling scheme (see chapter 3) has been implemented for some nonlinear process and resulted in a TS model of the form (5.1) with the following three local affine systems:

$$A_1 = \begin{bmatrix} -10 & -10 \\ 10 & 5 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -1 & -2 \\ 2 & -8 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} -10 & -11 \\ 10 & 9 \end{bmatrix}, \qquad B_3 = \begin{bmatrix} 0 \\ -2 \end{bmatrix}.$$

The weight functions $w_i(t)$ and the partitioning of the state-space are shown in Figure 5.1.

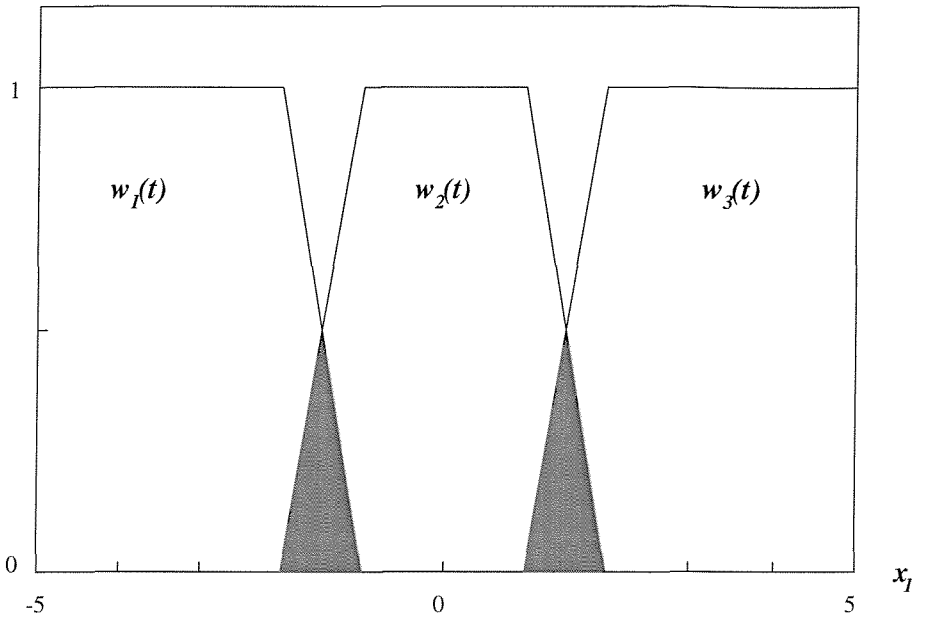

Figure 5.1: The weight functions and the state space partitioned into operating regions for example 5.1.

Therefore the global model of the system is

$$\dot{\mathbf{x}} = \sum_{i=1}^{3} w_i(t)[A_i\mathbf{x}(t) + B_i].$$

We note that the traditional method to find a common quadratic Lyapunov function for all local models is inapplicable here, because, according to theorem 4.3, the following dual problem can be solved: There exist positive semidefinite matrices R_i, not all zero, such that

$$\sum_{i=1}^{3} \bar{A}_i^T R_i + R_i \bar{A}_i \geq 0,$$

where $\bar{A}_i = \begin{bmatrix} A_i & B_i \\ 0 & 0 \end{bmatrix}$, $i = 1,2,3.$

Stating the stability conditions in theorem 5.3 for the partitioning in Figure 5.1,

we formulate (3+3+4) LMIs as follows:

$$
\begin{aligned}
\alpha I + \sum_k \mu_{1k}\bar{Q}_{1k} \le \bar{P}_1 \le \beta I - \sum_k \mu_{1k}\bar{Q}_{1k},\\
\alpha I + \sum_k \mu_{2k}\bar{Q}_{2k} \le \bar{P}_2 \le \beta I - \sum_k \mu_{2k}\bar{Q}_{2k},\\
\alpha I + \sum_k \mu_{3k}\bar{Q}_{3k} \le \bar{P}_3 \le \beta I - \sum_k \mu_{3k}\bar{Q}_{3k},\\
\bar{A}_1^T\bar{P}_1 + \bar{P}_1\bar{A}_1 + 2\bar{P}_1\sum_{j=1}^{3}\xi_{1j} + \sum_k \nu_{1k}\bar{Q}_{1k} \le -\gamma I,\\
\bar{A}_2^T\bar{P}_2 + \bar{P}_2\bar{A}_2 + 2\bar{P}_2\sum_{j=1}^{3}\xi_{2j} + \sum_k \nu_{2k}\bar{Q}_{2k} \le -\gamma I,\\
\bar{A}_3^T\bar{P}_3 + \bar{P}_3\bar{A}_3 + 2\bar{P}_3\sum_{j=1}^{3}\xi_{3j} + \sum_k \nu_{3k}\bar{Q}_{3k} \le -\gamma I,\\
\bar{P}_2 \le \bar{P}_1 - \eta_{12}\bar{R}_{12},\\
\bar{P}_1 \le \bar{P}_2 - \eta_{21}\bar{R}_{21},\\
\bar{P}_3 \le \bar{P}_2 - \eta_{23}\bar{R}_{23},\\
\bar{P}_2 \le \bar{P}_3 - \eta_{32}\bar{R}_{32},
\end{aligned}
\tag{5.14}
$$

Note that $\bar{Q}_{ik}$ and $\bar{R}_{ij}$ can be calculated according to (5.9) and (5.10) where Z_i and Λ_{ij} are local areas and boundaries of the partitioned operating regions shown in figure 5.1. A solution of the LMI system (5.14 is obtained via the routines of Matlab LMI library. The three obtained quadratic local Lyapunov functions are:

$$
\bar{P}_1 = \begin{bmatrix} 0.6042 & 0.4660 & 0.0553 \\ 0.4660 & 0.6733 & -0.0829 \\ 0.0553 & -0.0829 & 0.0553 \end{bmatrix},
$$

$$
\bar{P}_2 = \begin{bmatrix} 0.0143 & -0.0454 & 0.0000 \\ -0.0454 & 0.1447 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix},
$$

$$
\bar{P}_3 = 10^{-15}\begin{bmatrix} 0.2678 & 0.2043 & 0.0779 \\ 0.2043 & 0.1818 & 0.1105 \\ 0.0779 & 0.1105 & 0.0125 \end{bmatrix}.
$$

With the methods presented in Johansson and Rantzer (1998) and Pettersson and Lennartson (1997b), we need to solve (5+7) LMIs and (3+7+4) LMIs respectively. The computational difference between the proposed approach and those of Johansson and Rantzer (1998) and Pettersson and Lennartson (1997b) becomes large if the interpolation is performed in multiple dimensions for dynamical processes whose models

require high input dimension. Assume that fuzzy interpolation is also applied in the x_2-dimension in the example using the same membership function shape and position as in the x_1-dimension. Then we obtain 9 operating regions each with its own local dynamics, and 16 interpolation regions. In 4 of the interpolation regions 4 local dynamics are involved, and in the remaining 12 two local dynamics are involved. The total numbers of LMIs that need to be simultaneously solved are 25+((9*1)+(12*2)+(4*4))=74 and 9+((9*1)+(6*4)+(4*4))+24=82 with the methods presented in Johansson and Rantzer (1998) and Pettersson and Lennartson (1997b) respectively. But here we need only to solve 9+9+24=42 LMIs with our method. In the cast of three dimensional partitioning as Figure 5.2, the number of LMIs of the three methods are 468, 478, and 135 respectively, as illustrated in Figure 5.3.

Generally speaking, suppose the n-dimensional input space is decomposed by partitioning each axis with the same number of local areas, if each axis is partitioned into k areas, using both methods of Johansson and Rantzer (1998) and Pettersson and Lennartson (1997b) will need to solve $c_1[2(k-1)]^n$ number of LMIs, but using the proposed method only needs to solve $c_2 n^2 k^2$ number of LMIs, where c_1 and c_2 are polinomial functions of n.

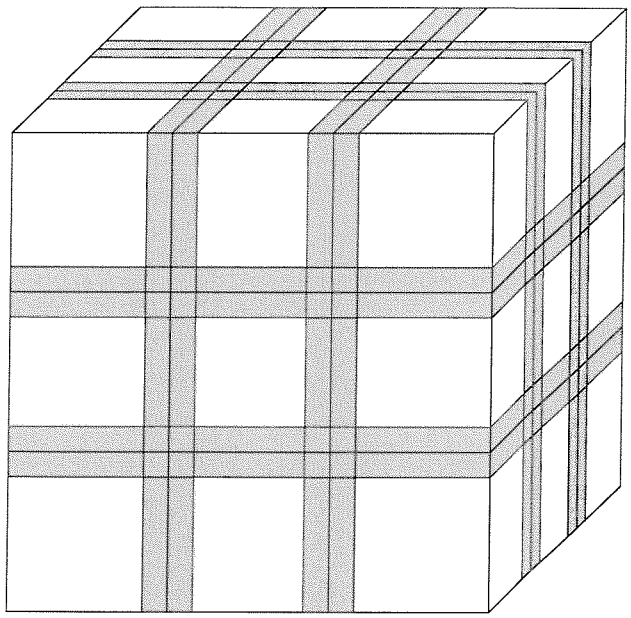

Figure 5.2: Three dimensional partitioning of the state space

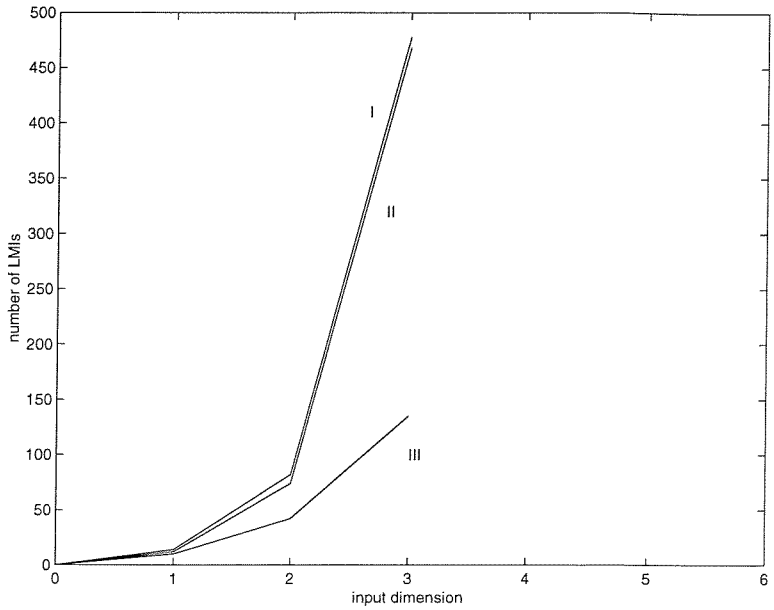

Figure 5.3: Comparison of the numbers of LMIs using three approaches: method I—Johansson; method II—Pettersson; method III—this chapter.

The numbers of LMIs using three approaches

5.4.2 Example 2

Also the approach presented in section 5.2 can be used for wider range of systems than that of Johansson and Rantzer (1998) and Pettersson and Lennartson (1997b) because the conditions are less conservative. The following example will illustrate the point. Consider (5.1) with the following three local affine systems:

$$A_1 = \begin{bmatrix} -10 & -10.5 \\ 10.5 & -9 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 11 \\ 7.5 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} -5 & -2.5 \\ 1 & -5 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} -10 & -10.5 \\ 10.5 & -10 \end{bmatrix}, \qquad B_3 = \begin{bmatrix} 11 \\ 10.5 \end{bmatrix},$$

The weight functions $w_i(t)$ and the partitioning of the state-space are shown in Figure 5.4.

Utilising the stability conditions of theorem 5.3 for the partitioning given in Figure 5.4, we formulate (3+3+4) LMIs. A solution of the LMI system is evaluated by the

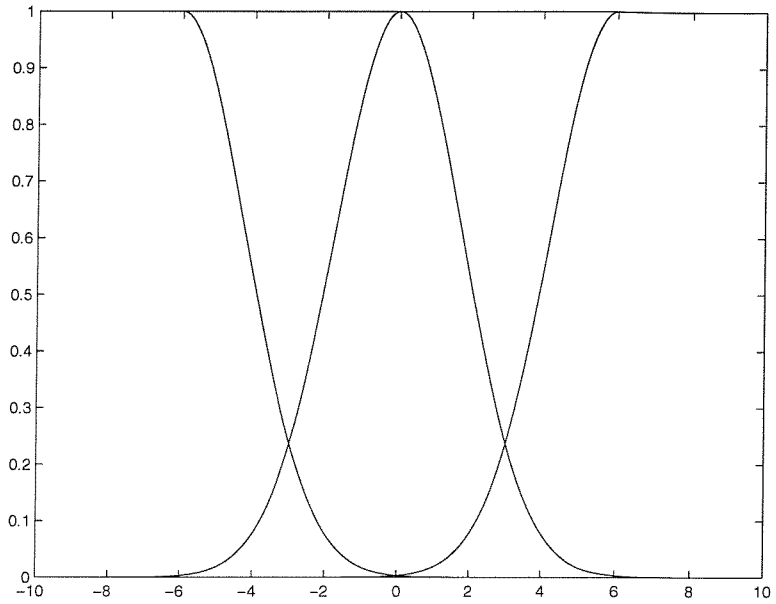

Figure 5.4: The weight functions and the state space partitioned into operating regions of example 5.2.

LMI toolbox of Matlab. The three obtained quadratic local Lyapunov functions are:

$$P_1 = 10^5 \begin{bmatrix} 1.3897 & -0.1854 & 0.0359 \\ -0.1854 & 1.2954 & -1.2136 \\ 0.0359 & -1.2136 & 1.1509 \end{bmatrix},$$

$$P_2 = 10^6 \begin{bmatrix} 0.7165 & -0.1585 & 0 \\ -0.1585 & 1.3928 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$P_3 = 10^5 \begin{bmatrix} 1.7358 & -0.1659 & 0.1760 \\ -0.1659 & 1.6024 & -1.6807 \\ 0.1760 & -1.6807 & 1.7629 \end{bmatrix}.$$

However, the LMIs formulated by methods presented in Johansson and Rantzer (1998) or Pettersson and Lennartson (1997b) are non-feasible using Matlab routines, as there is no common quadratic Lypunov function for the local models in the interpolation areas.

5.5 Concluding remarks

In this chapter, a new method for stability analysis of fuzzy systems is proposed. Similar to Pettersson and Lennartson (1997b), we provide a constructive technique to

seek a form of discontinuous Lyapunov functions, but, by making use of membership functions, this method needs only to find one local Lyapunov function for every interpolation region. Therefore the proposed technique drastically reduces the complexity of the associated LMI problem. Furthermore, because of the much simpler formulations in theorem 5.1 and theorem 5.2, we have relaxed the stability conditions of fuzzy systems given in Johansson and Rantzer (1998), and Pettersson and Lennartson (1997b) (which are considered the best stability results for fuzzy system stability to date).

Future research can be conducted along two directions. One is to integrate the stability determination with the modelling schemes described in chapter 3. Another one is to simplify the boundary conditions for local Lyapunov functions. We will discuss these ideas in more detail in chapter 8.

Next chapter will use the results obtained in this chapter to study control problems of local systems.

Chapter 6

Feedback stabilizing control of fuzzy systems

Various methods of fuzzy-model-based control have emerged as a powerful approach to the control design of complex non-linear systems (Takagi and Sugeno, 1985; Tanaka and Sugeno, 1992; Wang, 1996; Tanaka, Ikeda and Wang, 1996; Wang, Tanaka and Griffin, 1996), several of which have addressed this problem via the stabilization of state-space fuzzy models. Often, these solutions depend on finding a common Lyapunov function for the feedback control of nonlinear systems represented by a set of linear local models (Wang, Tanaka and Griffin, 1996; Kiriakidis, 1999). In Wang, Tanaka and Griffin (1996), a nonlinear system is represented by a set of TS models, and the control design is carried out by the so called *parallel distributed compensation scheme*. The idea is that for each local linear model, a linear feedback controller is designed. The resulting overall controller, which in general is nonlinear, is like the modelling representation, a fuzzy blending of each individual linear controller. Kiriakidis (1999) developed a framework that exploits the property of the fuzzy model as a convex hull of linear system matrices. Using a quasi-linear model structure, the robust stabilization of complex non-linear systems, against modelling error and parametric uncertainty, based on feedback control, is transformed into an LMI problem. In Cao, Rees and Feng (1997b), the authors presented a fuzzy controller design method which attempts to combine individual local linear based solutions to obtain a global solution for the overall design problem. The approach is based on the so called fuzzy dynamic model which is an extension of TS model. Instead of searching for a common Lyapunov function for all the subsystems, this algorithm sought for piecewise local quadratic Lyapunov functions. As such it generates a less conservative controller solution than that obtained in Kiriakidis (1999), although some boundary conditions

need to be incorporated. Another drawback of Cao, Rees and Feng (1997b) is that the resultant controller design is formulated so as to require a set of Riccati equations to be solved with an associated computational overhead.

In this chapter, following the results obtained in the preceding chapter, we formulate and solve the problem of robust stabilization for a broad class of fuzzy systems. Initially a class of affine continuous nonlinear dynamical systems are represented by a class of fuzzy models with sector bound modelling errors, for which the state space is partitioned into a set of local linear fuzzy models with bounded additive parametric and modelling errors/disturbances. In Section 6.2, state feedback controller design is derived by seeking a piecewise Lyapunov function for the closed-loop system so that the design solutions are minimally less conservative. Section 6.3 discusses output feedback controller design methods. An illustrative example is given in Section 6.4. Finally, section 6.5 concludes the chapter.

6.1 Fuzzy model of nonlinear systems

Here we consider the affine nonlinear dynamical systems described by (2.6), which is re-written here for convenience:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad x \in \Re^n, u \in \Re^m, \\ \mathbf{y} &= C\mathbf{x}, \end{aligned} \tag{6.1}$$

where the Jacobi matrix $\nabla\mathbf{f}(\mathbf{x})$ continuously differentiable and $\mathbf{f}(\mathbf{0}) = \mathbf{0}$, $\mathbf{g}(\mathbf{x}) \in \Re^{n\times m}$. Expanding $\mathbf{f}(\mathbf{x})$ about the origin we obtain $\mathbf{f}(\mathbf{x}) = A_s(\mathbf{x})\mathbf{x}$. Therefore system (6.1) can be approximated about the origin as

$$\dot{\mathbf{x}} = A_s(\mathbf{x})\mathbf{x} + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad \mathbf{x} \in \Re^n, \mathbf{u} \in \Re^m \tag{6.2}$$

Inside an arbitrarily large compact subspace $\mathcal{X} \in \Re^n$, the above class of non-linear systems can be approximated by a class of fuzzy models to any desired accuracy (Wang and Mendel, 1992). The fuzzy model that approximates the system matrix, $A_s(\mathbf{x})$, for all $\mathbf{x} \in \mathcal{X}$, stems from the following expansion in terms of *fuzzy basis functions* (FBFs) (Wang and Mendel, 1992; Kiriakidis, 1996)

$$A_s(\mathbf{x}) = \sum_{i=1}^{L} w_i(\mathbf{x})A_i + \Delta A(\mathbf{x}), \ A_i, \ \Delta A(\mathbf{x}) \in \Re^{n\times n}, \ \|\Delta A(\mathbf{x})\|_a \leq \bar{\mu}_A \tag{6.3}$$

where $\bar{\mu}_A > 0$ is arbitrarily small, L is finite, and the absolute matrix norm $\|D\|_a = n * \max_{1\le k,l\le n} |d_{kl}|$, $D \in \Re^{n\times n}$. The normalized fuzzy basis functions are given by

$$w_i(\mathbf{x}) = \frac{\bar{w}_i(\mathbf{x})}{\sum_{j=1}^{L} \bar{w}_j(\mathbf{x})}, \quad i = 1,...,L. \tag{6.4}$$

where $\bar{w}_i : \mathcal{X} \to [0,\ 1]$ are the fuzzy membership functions. Suppose that $n \le m$, if we set $B_s(\mathbf{x}) = [g(\mathbf{x})\ \mathbf{0}_{n\times(n-m)}]$, then similar to the process of representing $A_s(\mathbf{x})$ in FBFs we get

$$B_s(\mathbf{x}) = \sum_{i=1}^{L} w_i(\mathbf{x}) B_{si} + \Delta B_s(\mathbf{x}),\ \ B_{si},\ \Delta B_s(\mathbf{x}) \in \Re^{n\times n},\ \ \|\Delta B_s(\mathbf{x})\|_a \le \bar{\mu}_B \tag{6.5}$$

If we denote $B_{si} = [B_i\ \mathbf{0}_{n\times(n-m)}]$ and $\Delta B_s(\mathbf{x}) = [\Delta B\ \mathbf{0}_{n\times(n-m)}]$, then the resulting fuzzy system of (6.2) takes the form

$$\dot{\mathbf{x}} = A(\mathbf{x})\mathbf{x} + B(\mathbf{x})\mathbf{u} + \Delta\mathbf{f}(\mathbf{x},\mathbf{u}), \quad \mathbf{x} \in \mathcal{X}, \tag{6.6}$$

where

$$A(\mathbf{x}) = \sum_{i=1}^{L} w_i(\mathbf{x}) A_i, \quad B(\mathbf{x}) = \sum_{i=1}^{L} w_i(\mathbf{x}) B_i. \tag{6.7}$$

In the following it is assumed that the modelling error satisfies the sector bounds

$$\Delta\mathbf{f}^T(\mathbf{x},\mathbf{u})\Delta\mathbf{f}(\mathbf{x},\mathbf{u}) \le \mu^2 \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}^T \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}, \tag{6.8}$$

where μ is a constant. Note that the modelling error $\Delta\mathbf{f}(\mathbf{x},\mathbf{u})$ has been added into (2.21) to form (6.6) since (6.6) is considered to represent the original plant.

The fuzzy model description (6.6) is general enough to also incorporate perturbations that originate from other sources such as parameter uncertainty or external perturbations (Kiriakidis, 1999). For example, suppose that the matrix $A_s(\mathbf{x})$ in (6.2) depends linearly on a parameter vector θ, which is normal for say B-spline expansions of a general function $\mathbf{f}(\cdot)$. Then we have $A_s(\mathbf{x},\theta) = A_s(\mathbf{x},\theta_0) + \Delta A_p(\mathbf{x},\Delta\theta)$. We need only approximate the nominal part $A_s(\mathbf{x},\theta_0)$ by the fuzzy model as shown above, and obtain the following

$$A_s(\mathbf{x},\theta) = A(\mathbf{x},\theta_0) + \Delta A(\mathbf{x},\theta_0) + \Delta A_p(\mathbf{x},\Delta\theta)$$

Similarly $\mathbf{g}(\mathbf{x},\theta) = B(\mathbf{x},\theta_0) + \Delta B(\mathbf{x},\theta_0) + \Delta B_p(\mathbf{x},\Delta\theta)$ in (6.2). Here $\Delta A_p(\mathbf{x},\Delta\theta)$ and $\Delta B_p(\mathbf{x},\Delta\theta)$ represent parametric uncertainty, whereas $\Delta A(\mathbf{x},\theta_0)$ and $\Delta B(\mathbf{x},\theta_0)$ represent modelling error which are uniformly bounded as in (6.3) and (6.5). If the matrix

norms $\|\Delta A_p(\mathbf{x},\Delta\theta)\|_a$, $\|\Delta B_p(\mathbf{x},\Delta\theta)\|_a$ are also uniformly bounded, and their upper bounds are same as the upper bounds of $\|\Delta A(\mathbf{x},\theta_0)\|_a$ and $\|\Delta B(\mathbf{x},\theta_0)\|_a$, respectively, then the parametric uncertainty can be lumped with the modelling error.

Within each subregion Z_i the fuzzy system (6.6) can be denoted as

$$\begin{aligned}\dot{\mathbf{x}} &= (A_i+\Delta A_i)\mathbf{x}+(B_i+\Delta B_i)\mathbf{u}+\Delta\mathbf{f}(\mathbf{x},\mathbf{u}) \\ &= (A_i\mathbf{x}+B_i\mathbf{u})+(\Delta A_i\mathbf{x}+\Delta B_i\mathbf{u})+\Delta\mathbf{f}(\mathbf{x},\mathbf{u}) \quad \mathbf{x}\in Z_i \end{aligned} \tag{6.9}$$

where $\Delta A_i=\sum_{j=1}^{L} w_j(A_j-A_i)$ and $\Delta B_i=\sum_{j=1}^{L} w_j(B_j-B_i)$ $i=1,...,L$. We need the following important upper bounds of the system (6.9):

$$[\Delta A_i\ \Delta B_i]^T[\Delta A_i\ \Delta B_i]\le E_i^T E_i \tag{6.10}$$

where $E_i=[E_{i1}\ E_{i2}]$ is a constant matrix. If $\mathbf{u}\in\Re^m$, then $E_i\in\Re^{n\times(n+m)}$. The upper bounds E_i have to be selected by the designer. If unavailable, then the following search algorithm (Cao, Rees and Feng, 1997b) can be used to obtain approximate bounds from observable input/output data of the process (6.1):

Step 1. Denote $\bar{\Delta}A_i=\frac{1}{2}\sum_{j=1}^{L}(A_j-A_i)$ and $\bar{\Delta}B_i=\frac{1}{2}\sum_{j=1}^{L}(B_j-B_i)$, $i=1,...,L$. Choose the the approximate upper bounds (6.10) with the form:

$$E_i^T E_i=\lambda_i[\bar{\Delta}A_i\ \bar{\Delta}B_i]^T[\bar{\Delta}A_i\ \bar{\Delta}B_i]\ i=1,2,...L, \tag{6.11}$$

or the form

$$E_i^T E_i=\lambda_i I,\ i=1,2,...,L. \tag{6.12}$$

Step 2. Initially, set $\lambda_i=0$, $i=1,2,...,L$, and determine whether the closed-loop subsystems in (6.14) are stable with $\Delta\mathbf{f}(\mathbf{x})=0$. If any one does not, the algorithm ends and the upper bounds cannot be found; otherwise go to Step 3.

Step 3. Increase the values of $\lambda_i=\lambda_i+\sigma_i$, where $\sigma_i>0, i=1,2,...,L$. Then determine whether the set of closed-loop subsystems in (6.14) are stable. If they are all stable then repeat Step 3 until the maximum values of λ_i, $i=1,2,...,L$ have been found, which guarantee that the set of closed-loop subsystems (6.14) are stable. Substituting These values into (6.11) we obtained the approximate upper bounds.

Remark: It should be noted that the above searching algorithm gives the largest allowable upper bounds, which represent the largest allowable interaction among the subsystems.

6.2 State feedback controller

Initially we assume that all states, $\mathbf{x}$, are available from observation. Later in Section 6.3 we relax this assumption. In this case, a piecewise static state feedback controller of the form

$$\mathbf{u} = K_i\mathbf{x}, \quad \mathbf{x} \in Z_i \tag{6.13}$$

can be used for system (6.9) to form a closed-loop system

$$\dot{\mathbf{x}} = (A_i + B_iK_i)\mathbf{x} + (\Delta A_i + \Delta B_iK_i)\mathbf{x} + \Delta\mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in Z_i, \tag{6.14}$$

where $\Delta\mathbf{f}(\mathbf{x})$ is a simplified notation for $\Delta\mathbf{f}(\mathbf{x}, K_i\mathbf{x})$ and satisfies the sector bounds

$$\Delta\mathbf{f}^T(\mathbf{x})\Delta\mathbf{f}(\mathbf{x}) \leq \mu^2\mathbf{x}^T \begin{bmatrix} I \\ K_i \end{bmatrix}^T \begin{bmatrix} I \\ K_i \end{bmatrix} \mathbf{x}, \tag{6.15}$$

where I is an identity matrix with suitable dimension.

From the upper bound of (6.10), we also have

$$(\Delta A_i + \Delta B_iK_i)^T(\Delta A_i + \Delta B_iK_i) \leq \begin{bmatrix} I \\ K_i \end{bmatrix}^T E_i^T E_i \begin{bmatrix} I \\ K_i \end{bmatrix} \tag{6.16}$$

Applying theorem 4.4 we can prove the following lemma (Feng and Harris, 1999b):

Lemma 6.1 *Assume that the state trajectory evolves according to (6.6). If there exist quadratic functions $V_i(\mathbf{x}) = \mathbf{x}P_i\mathbf{x}$, $i \in I_L$, such that*

(i) $\forall \mathbf{x} \neq 0 \in Z_i, V_i(\mathbf{x}) > 0, i = 1, \ldots L,$

(ii) $\forall \mathbf{x} \in Z_i, \dot{V}_i(\mathbf{x}) < 0, i = 1, \ldots L,$

(iii) $\forall \mathbf{x} \in \Lambda_{ij}, V_j(\mathbf{x}) \leq V_i(\mathbf{x}), (i,j) \in I_\Lambda,$

then the system is (uniformly) exponentially stable in the sense of Lyapunov.

Proof Let

$$\alpha = \min_i \lambda_{min}(P_i), \qquad \beta = \max_i \lambda_{max}(P_i),$$

then for any $i = \in I_L$, within the region $\mathbf{x} \in Z_i$,

$$\alpha\|\mathbf{x}\|^2 \leq V_i(\mathbf{x}) \leq \beta\|\mathbf{x}\|^2.$$

Also, Solutions to the strict inequalities in (**ii**) implies the existence of positive real numbers $\gamma_i > 0$ such that

$$\dot{V}_i(\mathbf{x}) \leq -\gamma_i \|\mathbf{x}\|^2.$$

Denote $\gamma = \min_i \gamma_i$, then

$$\dot{V}_i(\mathbf{x}) \leq -\gamma \|\mathbf{x}\|^2.$$

Now it is direct from theorem 4.4 that the lemma can be established.

Using lemma 6.1, we obtain the following stability result for the closed-loop system (6.14) (Feng and Harris, 2000):

Theorem 6.1 *Suppose there exist positive definite matrices* $\{Q_1, ..., Q_L\}$, *matrices* $\{Y_1, ..., Y_L\}$, R_{ij}, *and constants* η_{ij} *such that the following conditions hold for* $i = 1, ..., L$

(i)

$$\begin{bmatrix} Q_i A_i^T + A_i Q_i + B_i Y_i + Y_i^T B_i^T + 2I & \begin{bmatrix} Q_i & Y_i^T \end{bmatrix} \\ \begin{bmatrix} Q_i \\ Y_i \end{bmatrix} & -(\mu^2 I - E_i^T E_i)^{-1} \end{bmatrix} < 0; \tag{6.17}$$

(ii)

$$Q_i \leq Q_j - \eta_{ij} R_{ij}, \qquad (i, j) \in I_\Lambda. \tag{6.18}$$

Then the closed-loop (6.14), subject to (6.10) and (6.15), with feedback gain $K_i = Y_i Q_i^{-1}$, $i = 1, ..., L$ *is stable in the sense of Lyapunov.*

Proof According to Lemma 6.1, we need only to find a piecewise Lyapunov function for the system to ensure the stability. Consider the Lyapunov function candidate given by $V_i = \mathbf{x}^T P_i \mathbf{x}$, where $P_i = Q_i^{-1}$. V_i is positive for $i = 1, ..., L$. In the following we will prove that $\dot{V}_i < 0$ for all $\mathbf{x} \neq 0$ and $i = 1, ..., L$, or equivalently

$$\begin{aligned} \dot{V}_i &= \mathbf{x}^T (A_i + B_i K_i)^T P_i \mathbf{x} + \mathbf{x}^T P_i (A_i + B_i K_i) \mathbf{x} \\ &\quad + \mathbf{x}^T (\Delta A_i + \Delta B_i K_i)^T P_i \mathbf{x} + \mathbf{x}^T P_i (\Delta A_i + \Delta B_i K_i) \mathbf{x} + \Delta \mathbf{f}(\mathbf{x})^T P_i \mathbf{x} + \mathbf{x}^T P_i \Delta \mathbf{f}(\mathbf{x}) \\ &= \bar{\mathbf{x}}_i^T \bar{A}_i \bar{\mathbf{x}}_i < 0, \end{aligned} \tag{6.19}$$

where we denote

$$\bar{\mathbf{x}}_i = \begin{bmatrix} \mathbf{x} \\ (\Delta A_i + \Delta B_i K_i)\mathbf{x} \\ \Delta \mathbf{f}(\mathbf{x}) \end{bmatrix}, \qquad \bar{A}_i = \begin{bmatrix} (A_i + B_i K_i)^T P_i + P_i (A_i + B_i K_i) & P_i & P_i \\ P_i & 0 & 0 \\ P_i & 0 & 0 \end{bmatrix}.$$

By (6.15) and (6.16), $\Delta A_i, \Delta B_i$ and $\Delta f(x)$ satisfy the following condition

$$\bar{\mathbf{x}}_i^T \begin{bmatrix} -\mu^2 (I + K_i^T K_i) - (E_{i1} + E_{i2} K_i)^T (E_{i1} + E_{i2} K_i) & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \bar{\mathbf{x}}_i \leq 0 \tag{6.20}$$

Applying the $\mathcal{S}$-procedure, inequalities (6.19) and (6.20) hold if there exists $\tau \geq 0$ such that

$$\bar{A}_i - \tau \begin{bmatrix} -\mu^2 (I + K_i^T K_i) - (E_{i1} + E_{i2} K_i)^T (E_{i1} + E_{i2} K_i) & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} < 0. \tag{6.21}$$

From the Schur complement lemma 4.1, inequality (6.21) hold if and only if there exists $\tau \geq 0$ such that

$$\begin{aligned} &(A_i + B_i K_i)^T P_i + P_i (A_i + B_i K_i) + \tau (\mu^2 (I + K_i^T K_i) + (E_{i1} + E_{i2} K_i)^T (E_{i1} + E_{i2} K_i)) \\ &- [P_i \; P_i] \begin{bmatrix} -\tau^{-1} I & 0 \\ 0 & -\tau^{-1} I \end{bmatrix} \begin{bmatrix} P_i \\ P_i \end{bmatrix} < 0. \end{aligned} \tag{6.22}$$

Substituting $Q_i = \tau P_i^{-1}$ and $Y_i = K_i Q_i$, after manipulation, inequality (6.22) becomes

$$Q_i A_i^T + A_i Q_i + B_i Y_i + Y_i^T B_i^T + 2I + \begin{bmatrix} Q_i \\ Y_i \end{bmatrix}^T (\mu^2 I + E_i^T E_i) \begin{bmatrix} Q_i \\ Y_i \end{bmatrix} < 0. \tag{6.23}$$

Using the Schur complement lemma on (6.17), (6.23) follows, and so the first and second conditions of Lemma 6.1 hold. Since $P_i \geq P_j$ if and only if $Q_i \leq Q_j$, the third condition of Lemma 6.1 is a direct consequence of (6.18) after applying the $\mathcal{S}$-procedure, establishing the theorem.

Remark: The conditions of theorem 6.1 is a set of LMIs and so it is easy to use standard software packages to find their solution. Therefore it provides an easy construction method for controller design.

6.3 Output feedback controller design

The above state feedback control requires that the system states are available. In practice this requirement may be too restrictive since frequently only an input/output

model is available from online observations. In this section we consider the control law based on output feedback. Recall in (6.1) the output equation for the fuzzy system is

$$\mathbf{y} = C\mathbf{x}, \tag{6.24}$$

based on which we construct a state observer and output feedback controller via

$$\dot{\mathbf{z}} = A_c\mathbf{z} + B_c\mathbf{y}, \tag{6.25}$$

$$\mathbf{u} = K_i\mathbf{z}, \quad \mathbf{z} \in Z_i, \tag{6.26}$$

where A_c, B_c and K_i are controller matrices to be determined. Applying the output feedback controller (6.26) to system (6.9), the closed loop fuzzy control system can be obtained as

$$\dot{\mathbf{x}} = (A_i\mathbf{x} + B_iK_i\mathbf{z}) + (\Delta A_i\mathbf{x} + \Delta B_iK_i\mathbf{z}) + \Delta\mathbf{f}(\mathbf{x},\mathbf{z}) \tag{6.27}$$

Define the augment state vector $\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} \in \Re^{2n}$, then (6.25) and (6.27) becomes on using (6.24)

$$\dot{\tilde{\mathbf{x}}} = \tilde{A}_i\tilde{\mathbf{x}} + \Delta\tilde{A}_i(\tilde{\mathbf{x}})\tilde{\mathbf{x}} + \Delta\tilde{\mathbf{f}}(\tilde{\mathbf{x}}), \tag{6.28}$$

where

$$\tilde{A}_i = \begin{bmatrix} A_i & B_iK_i \\ B_cC & A_c \end{bmatrix}, \quad \Delta\tilde{A}_i(\tilde{\mathbf{x}}) = \begin{bmatrix} \Delta A_i & \Delta B_iK_i \\ 0 & 0 \end{bmatrix}, \quad \Delta\tilde{\mathbf{f}}(\tilde{\mathbf{x}}) = \begin{bmatrix} \Delta\mathbf{f}(\mathbf{x},\mathbf{z}) \\ 0 \end{bmatrix}.$$

From (6.8) and (6.10) the following inequalities hold, respectively

$$\tilde{\mathbf{x}}^T\Delta\tilde{A}_i^T(\tilde{\mathbf{x}})\Delta\tilde{A}_i(\tilde{\mathbf{x}})\tilde{\mathbf{x}} \leq \tilde{\mathbf{x}}^T\tilde{K}_i^TE_i^TE_i\tilde{K}_i\tilde{\mathbf{x}}, \tag{6.29}$$

$$\Delta\tilde{\mathbf{f}}^T(\tilde{\mathbf{x}})\Delta\tilde{\mathbf{f}}(\tilde{\mathbf{x}}) \leq \mu^2\tilde{\mathbf{x}}^T\tilde{K}_i^T\tilde{K}_i\tilde{\mathbf{x}}, \tag{6.30}$$

where $\tilde{K}_i = \begin{bmatrix} I & 0 \\ 0 & K_i \end{bmatrix}$. We have the following lemma:

Lemma 6.2 *Assume that the state trajectory evolves according to (6.6). If there exist matrices $\tilde{P}_i > 0$, and K_i such that*

$$\tilde{A}_i^T\tilde{P}_i + \tilde{P}_i\tilde{A}_i + \mu^2\tilde{K}_i^T\tilde{K}_i + \tilde{K}_i^TE_i^TE_i\tilde{K}_i + 2\tilde{P}_i^2 < 0,$$

and we apply the observer (6.25) and controllers (6.26) to (6.6), then the closed loop system, subject to (6.8) and (6.10), is exponentially stable in the sense of Lyapunov.

Proof: Initially consider the Lyapunov function candidate $V = \tilde{\mathbf{x}}^T \tilde{P}_i \tilde{\mathbf{x}}$, where $\tilde{P}_i$ is a positive matrix. We have

$$\dot{V} = \begin{bmatrix} \tilde{\mathbf{x}} \\ \Delta\tilde{A}_i\tilde{\mathbf{x}} \\ \Delta\mathbf{f} \end{bmatrix}^T \begin{bmatrix} \tilde{A}_i^T\tilde{P}_i + \tilde{P}_i\tilde{A}_i & \tilde{P}_i & \tilde{P}_i \\ \tilde{P}_i & 0 & 0 \\ \tilde{P}_i & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \Delta\tilde{A}_i\tilde{\mathbf{x}} \\ \Delta\mathbf{f} \end{bmatrix}. \tag{6.31}$$

From (6.29) and (6.30) we also have

$$\begin{bmatrix} \tilde{\mathbf{x}} \\ \Delta\tilde{A}_i\tilde{\mathbf{x}} \\ \Delta\mathbf{f} \end{bmatrix}^T \begin{bmatrix} -\mu^2\tilde{K}_i^T\tilde{K}_i - \tilde{K}_i^T E_i^T E_i \tilde{K}_i & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \Delta\tilde{A}_i\tilde{\mathbf{x}} \\ \Delta\mathbf{f} \end{bmatrix} < 0. \tag{6.32}$$

Applying the $\mathcal{S}$-procedure, we know that $\dot{V} < 0$ and (6.32) hold if there exists $\tau > 0$ such that

$$\begin{bmatrix} \tilde{A}_i^T\tilde{P}_i + \tilde{P}_i\tilde{A}_i & \tilde{P}_i & \tilde{P}_i \\ \tilde{P}_i & 0 & 0 \\ \tilde{P}_i & 0 & 0 \end{bmatrix} - \tau \begin{bmatrix} -\mu^2\tilde{K}_i^T\tilde{K}_i - \tilde{K}_i^T E_i^T E_i\tilde{K}_i & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} < 0.$$

From the Schur complement lemma 4.1, the above inequality is equivalent to

$$\tilde{A}_i^T\tilde{P}_i + \tilde{P}_i\tilde{A}_i + \tau\mu^2\tilde{K}_i^T\tilde{K}_i + \tau\tilde{K}_i^T E_i^T E_i\tilde{K}_i + 2\tau^{-1}\tilde{P}_i^2 < 0$$

After multiplying by τ^{-1}, and re-denote $\tau^{-1}\tilde{P}_i$ as $\tilde{P}_i$ we get

$$\tilde{A}_i^T\tilde{P}_i + \tilde{P}_i\tilde{A}_i + \mu^2\tilde{K}_i^T\tilde{K}_i + \tilde{K}_i^T E_i^T E_i\tilde{K}_i + 2\tilde{P}_i^2 < 0. \tag{6.33}$$

This proves the lemma.

If we use the following observers:

$$\dot{\mathbf{z}} = A_i\mathbf{z} + B_i\mathbf{y} \qquad \mathbf{z} \in Z_i, \tag{6.34}$$

then we obtain the following theorem from Lemma 6.2:

Theorem 6.2 *Assume that the state trajectory evolves according to (6.6). We apply the observers (6.34) and controllers (6.26) to (6.6). If there exist*

(i) *matrices $Q_i > 0$, Y_i, $i = 1,...L$ such that*

$$\begin{bmatrix} Q_iA_i^T + A_iQ_i + 2I & \mu Y_i^T & Y_i^T E_{i2}^T \\ \mu Y_i & -I & 0 \\ E_{i2}Y_i & 0 & -I \end{bmatrix} < 0, \tag{6.35}$$

(ii) *matrices $P_i > 0$, $i = 1, ...L$ such that*

$$\begin{bmatrix} M_i^T P_i + P_i M_i + \mu^2 I + E_{i1}^T E_{i1} - N_i & P_i \\ P_i & (B_i K_i Q_i G_i^{-1} Q_i (B_i K_i)^T - 2I)^{-1} \end{bmatrix} < 0, \tag{6.36}$$

where

$$\begin{aligned} K_i &= Y_i Q_i^{-1}, \\ G_i &= Q_i A_i^T + A_i Q_i + \mu^2 Y_i^T Y_i + Y_i^T E_{i2}^T E_{i2} Y_i + 2I, \\ M_i &= A_i - B_i K_i Q_i G_i^{-1} (B_i C + Y_i^T E_{i2}^T E_{i1}), \\ N_i &= (B_i C + Y_i^T E_{i2}^T E_{i1})^T G_i^{-1} (B_i C + Y_i^T E_{i2}^T E_{i1}). \end{aligned} \tag{6.37}$$

(iii) *matrices R_{ij}, and constants η_{ij}, $i, j = 1, ..., L$ such that*

$$\begin{bmatrix} P_i & \\ & Q_i^{-1} \end{bmatrix} \leq \begin{bmatrix} P_j & \\ & Q_j^{-1} \end{bmatrix} - \eta_{ij} R_{ij}, (i, j) \in I_\Lambda. \tag{6.38}$$

Then the closed loop system, subject to (6.8) and (6.10), with $K_i = Y_i Q_i^{-1}$, is stable in the sense of Lyapunov.

Proof: Let's consider the positive matrix $\tilde{P}_i = \begin{bmatrix} P_i & \\ & Q_i^{-1} \end{bmatrix}$ where P_i and Q_i are $n \times n$ positive matrices. Set $Y_i = K_i Q_i$. Then $\tilde{K}_i \tilde{P}_i^{-1} = \begin{bmatrix} P_i^{-1} & \\ & Y_i \end{bmatrix}$. Pre- and post-multiply (6.33) by $\tilde{P}_i^{-1}$, and after manipulation, we get

$$\begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} < 0$$

where

$$\begin{aligned} e_{11} &= P_i^{-1} A_i^T + A_i P_i^{-1} + \mu^2 P_i^{-2} + P_i^{-1} E_{i1}^T E_{i1} P_i^{-1} + 2I, \\ e_{12} &= P_i^{-1} (B_c C)^T + B_i K_i Q_i + P_i^{-1} E_{i1}^T E_{i2} Y_i, \\ e_{21} &= Q_i (B_i K_i)^T + B_c C P_i^{-1} + Y_i^T E_{i2}^T E_{i1} P_i^{-1}, \\ e_{22} &= Q_i A_c^T + A_c Q_i + \mu^2 Y_i^T Y_i + Y_i^T E_{i2}^T E_{i2} Y_i + 2I. \end{aligned}$$

Applying Schur complement lemma, and using the definition in (6.34) and (6.37), the above condition holds if

$$Q_i A_i^T + A_i Q_i + \mu^2 Y_i^T Y_i + Y_i^T E_{i2}^T E_{i2} Y_i + 2I < 0, \tag{6.39}$$

and

$$P_i^{-1}M_i^T + M_iP_i^{-1} + \mu^2 P_i^{-2} + P_i^{-1}E_{i1}^T E_{i1}P_i^{-1} \\ +2I - P_i^{-1}N_iP_i^{-1} - B_iK_iQ_iG_i^{-1}Q_i(B_iK_i)^T < 0. \tag{6.40}$$

Pre- and post-multiplying (6.40) by P_i, and applying Schur complement lemma again, (6.39) equivalent to (6.35), and (6.40) equivalent to (6.36). Therefore, (6.35) and (6.36) enable us to ensure $\dot{V} < 0$. Also, similar to the proof of theorem 6.1, the inequality (6.38) guarantees the third condition of Lemma 6.1 to be satisfied, establishing the theorem.

6.3.1 Observer design

To obtain an improved performance of the closed loop system, we can further use the LMIs technique to design the observers in (6.25). We summarize the method in the following theorem:

Theorem 6.3 *Assume that the state trajectory evolves according to (6.6). We apply the observers (6.25) and controllers (6.26) to (6.6). If for* $i = 1, ..., L$, *there exist matrices* X_i, Y_i, Γ_{i1}, Γ_{i2}, Γ_{i3} *and* Γ_{i4} *such that*

$$\begin{bmatrix} \bar{m}_{11} & \Gamma_{i2}+A_i & \mu Y_i & \mu\Gamma_{i1} & Y_iE_{1i}^T+\Gamma_{i1}E_{2i}^T & Y_i & I \\ \Gamma_{i2}^T+A_i^T & \bar{m}_{22} & \mu I & 0 & E_{1i}^T & I & X_i \\ \mu Y_i & \mu I & -I & 0 & 0 & 0 & 0 \\ \mu\Gamma_{i1}^T & 0 & 0 & -I & 0 & 0 & 0 \\ E_{1i}Y_i^T+E_{2i}\Gamma_{i1}^T & E_{1i} & 0 & 0 & -I & 0 & 0 \\ Y_i & I & 0 & 0 & 0 & -2\Gamma_{i4} & -2Y_i \\ I & X_i & 0 & 0 & 0 & -2Y_i & -2I \end{bmatrix} < 0, \tag{6.41}$$

where

$$\begin{aligned} \bar{m}_{11} &= Y_iA_i^T + A_iY_i + \Gamma_{i1}B_i^T + B_i\Gamma_{i1}^T, \\ \bar{m}_{22} &= A_i^TX_i + X_i^TA_i + C^T\Gamma_{i3}^T + \Gamma_{i3}C, \end{aligned}$$

and

$$\begin{bmatrix} Y_i & I \\ I & X_i \end{bmatrix} > 0, \tag{6.42}$$

and for $(i,j) \in I_\Lambda$, *there exist matrices* R_{ij}, H_i, H_j *and constants* η_{ij} *such that*

$$\begin{bmatrix} X_i & D_i \\ D_i^T & H_i \end{bmatrix} < \begin{bmatrix} X_j & D_j \\ D_j^T & H_j \end{bmatrix} - \eta_{ij} R_{ij}, \tag{6.43}$$

where D_i *and* F_i *are obtained from*

$$D_i F_i^T = I - X_i Y_i,$$

then the closed loop system, subject to (6.8) and (6.10), is exponentially stable in the sense of Lyapunov. Furthermore, the observer and controller can be calculated as follows

$$\begin{aligned} K_i &= \Gamma_{i1}^T (F_i^{-1})^T, \\ B_c &= D_i^{-1} \Gamma_{i3}, \\ A_c &= D_i^{-1} (\Gamma_{i2}^T - X_i A_i Y_i - X_i B_i \Gamma_{i1}^T - \Gamma_{i3} C Y_i)(F_i^{-1})^T, \end{aligned} \tag{6.44}$$

for $\mathbf{z} \in Z_i$.

Proof: According to lemma 6.2, if we have matrices $\tilde{P}_i > 0$ and $\tilde{K}_i$ such that

$$\tilde{A}_i^T \tilde{P}_i + \tilde{P}_i \tilde{A}_i + \mu^2 \tilde{K}_i^T \tilde{K}_i + \tilde{K}_i^T E_i^T E_i \tilde{K}_i + 2\tilde{P}_i^2 < 0$$

then the closed-loop system is stable. Apply the Schur complement lemma to above inequality we have

$$\begin{bmatrix} \tilde{A}_i^T \tilde{P}_i + \tilde{P}_i \tilde{A}_i & \mu \tilde{K}_i^T & \tilde{K}_i^T E_i^T & \tilde{P}_i \\ \mu \tilde{K}_i & -I & 0 & 0 \\ E_i \tilde{K}_i & 0 & -I & 0 \\ \tilde{P}_i & 0 & 0 & -2I \end{bmatrix} < 0. \tag{6.45}$$

Now we consider non-diagonal positive matrix of the form

$$\tilde{P}_i = \begin{bmatrix} X_i & D_i \\ D_i^T & H_i \end{bmatrix}.$$

We denote

$$\tilde{P}_i^{-1} = \begin{bmatrix} Y_i & F_i \\ F_i^T & * \end{bmatrix}, \quad \tilde{W}_i = \begin{bmatrix} Y_i & I \\ F_i^T & 0 \end{bmatrix}.$$

Note that $X_iY_i + D_iF_i^T = I$, and $\tilde{P}_i\tilde{W}_i = \begin{bmatrix} I & X_i \\ 0 & D_i^T \end{bmatrix}$. suppose F_i is invertible, then (6.45) is equivalent to

$$\begin{bmatrix} \tilde{W}_i & & & \\ & I & & \\ & & I & \\ & & & \tilde{W}_i \end{bmatrix}^T \begin{bmatrix} \tilde{A}_i^T\tilde{P}_i + \tilde{P}_i\tilde{A}_i & \mu\tilde{K}_i^T & \tilde{K}_i^T E_i^T & \tilde{P}_i \\ \mu\tilde{K}_i & -I & 0 & 0 \\ E_i\tilde{K}_i & 0 & -I & 0 \\ \tilde{P}_i & 0 & 0 & -2I \end{bmatrix} \begin{bmatrix} \tilde{W}_i & & & \\ & I & & \\ & & I & \\ & & & \tilde{W}_i \end{bmatrix} < 0,$$

which in turn is equivalent to

$$\begin{bmatrix} m_{11} & m_{12} & \mu Y_i & \mu F_i K_i^T & m_{15} & Y_i & I \\ m_{12}^T & m_{22} & \mu I & 0 & E_{1i}^T & I & X_i \\ \mu Y_i & \mu I & -I & 0 & 0 & 0 & 0 \\ \mu K_i F_i^T & 0 & 0 & -I & 0 & 0 & 0 \\ m_{15}^T & E_{1i} & 0 & 0 & -I & 0 & 0 \\ Y_i & I & 0 & 0 & 0 & -2m_{66} & -2Y_i \\ I & X_i & 0 & 0 & 0 & -2Y_i & -2I \end{bmatrix} < 0, \tag{6.46}$$

where

$$\begin{aligned} m_{11} &= Y_iA_i^T + A_iY_i + F_iK_i^TB_i^T + B_iK_iF_i^T, \\ m_{12} &= Y_iA_i^TX_i + F_iK_i^TB_i^TX_i + Y_iC^TB_c^TD_i^T + F_iA_c^TD_i^T + A_i, \\ m_{15} &= Y_iE_{i1}^T + F_iK_i^TE_{i2}^T, \\ m_{22} &= A_i^TX_i + X_i^TA_i + C^TB_c^TD_i^T + D_iB_cC, \\ m_{66} &= Y_i^2 + F_iF_i^T. \end{aligned}$$

Define

$$\begin{aligned} \Gamma_{i1} &= F_iK_i^T, \\ \Gamma_{i2} &= Y_iA_i^TX_i + F_iK_i^TB_i^TX_i + Y_iC^TB_c^TD_i^T + F_iA_c^TD_i^T, \\ \Gamma_{i3} &= D_iB_c, \\ \Gamma_{i4} &= Y_i^2 + F_iF_i^T. \end{aligned} \tag{6.47}$$

Substituting (6.47) into (6.46) we obtain (6.41). Furthermore,

$$\tilde{W}_i^T\tilde{P}_i\tilde{W}_i = \begin{bmatrix} Y_i & I \\ I & X_i \end{bmatrix} > 0.$$

Therefore $\tilde{P}_i > 0$. Similarly to theorem 6.2, the inequality (6.43) guarantees that the third condition of lemma 6.1 is satisfied. (6.44) is a direct consequence of (6.47). The proof is complete.

6.4 Illustrative design example

Consider as an illustrative problem, the stabilization of a simple spacecraft. The dynamic behaviour of the spacecraft can be modelled by (Kiriakidis, 1999; Wang, Tanaka and Griffin, 1996)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{g\sin(x_1)-aml x_2^2\sin(2x_1)/2-a\cos(x_1)u}{4l/3-aml\cos^2(x_1)} \end{bmatrix}. \tag{6.48}$$

where x_1 is the angular displacement of the axis of symmetry from the upward vertical equilibrium, x_2 the angular velocity, u the force from the thrusters, and g the acceleration of gravity. For a planet the size of Earth, g is given by

$$g = g_0 + \Delta g = g_0\left(\frac{6370\times 10^3}{6370\times 10^3 + H}\right)^2,$$

where $g_0 = 9.8m/s^2$ and H the altitude of the spacecraft in kilometers from the surface of the planet. Taking for a typical micro spacecraft, $m = 2$kg, $M = 8$kg, $l = 1$m and $a = \frac{1}{m+M}$. The altitude variation of the acceleration of gravity induces parametric uncertainty in (6.48). We treat this uncertainty as a perturbation term. Denote $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. The nominal drift dynamics can be obtained from Wang, Tanaka and Griffin (1996). Combining the nominal drift dynamics and the uncertainty part produced by H the altitude, we obtain the following fuzzy model for system (6.48) in the subspace $\mathcal{X} = [-(\pi/2), \pi/2]\times(-\infty,\infty)$:

$$\dot{\mathbf{x}} = \sum_{i=1}^{2} w_i(\mathbf{x})(A_i\mathbf{x} + B_i u) + \Delta\mathbf{f}(\mathbf{x}), \tag{6.49}$$

where w_i are calculated according to (6.4), and

$$\bar{w}_1(\mathbf{x}) = \begin{cases} +\frac{2x_1}{\pi}, & -\frac{\pi}{2} \le x_1 \le 0, \\ 1-\frac{2x_1}{\pi}, & 0 \le x_1 \le \frac{\pi}{2}, \end{cases}$$

$$\bar{w}_2(\mathbf{x}) = \begin{cases} -\frac{2x_1}{\pi}, & -\frac{\pi}{2} \le x_1 \le 0, \\ \frac{2x_1}{\pi}, & 0 \le x_1 \le \frac{\pi}{2}. \end{cases}$$

The local models are

$$A_1 = \begin{bmatrix} 0 & 1 \\ \frac{g_0}{4l/3-aml} & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\ \frac{2g_0}{\pi(4l/3-aml\beta^2)} & 0 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0 \\ \frac{-a}{4l/3-aml} \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ \frac{-\beta a}{4l/3-aml\beta^2} \end{bmatrix},$$

$$\Delta\mathbf{f}(x) = \begin{bmatrix} 0 \\ \frac{\Delta g\sin(x_1)}{4l/3-aml\cos^2(x_1)} \end{bmatrix},$$

where $\beta = \cos(88^o)$. Note that the model (6.49) is different from the model of Wang, Tanaka and Griffin (1996) since we include the modelling error term $\Delta\mathbf{f}(\mathbf{x})$. For $x_1 \in [-(\pi/2), \pi/2]$ and $H \in [0, 100]$ km, we have $\|\Delta\mathbf{f}(\mathbf{x})\| \leq 6 \times 10^{-4}\|\mathbf{x}\|$, i.e. $\mu = 6 \times 10^{-4}$.

6.4.1 State feedback control

Solving the LMIs (6.17) and (6.18) formulated in theorem 6.1 results in the following matrices:

$$Q_1 = 10^3 \times \begin{bmatrix} 0.0292 & -0.0571 \\ -0.0571 & 1.8079 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 54.4335 & -144.6490 \\ -144.6490 & 393.9323 \end{bmatrix},$$

$$Y_1 = 10^4 \times \begin{bmatrix} 1.2161 \\ 3.6264 \end{bmatrix}, \quad Y_2 = 10^4 \times \begin{bmatrix} 2.2208 \\ -5.8098 \end{bmatrix}.$$

The response of the resultant state feedback control system with initial conditions $(x_1, x_2) = (15^o, 0)$ is plotted in Figure 6.1. The control input is plotted in Figure 6.2.

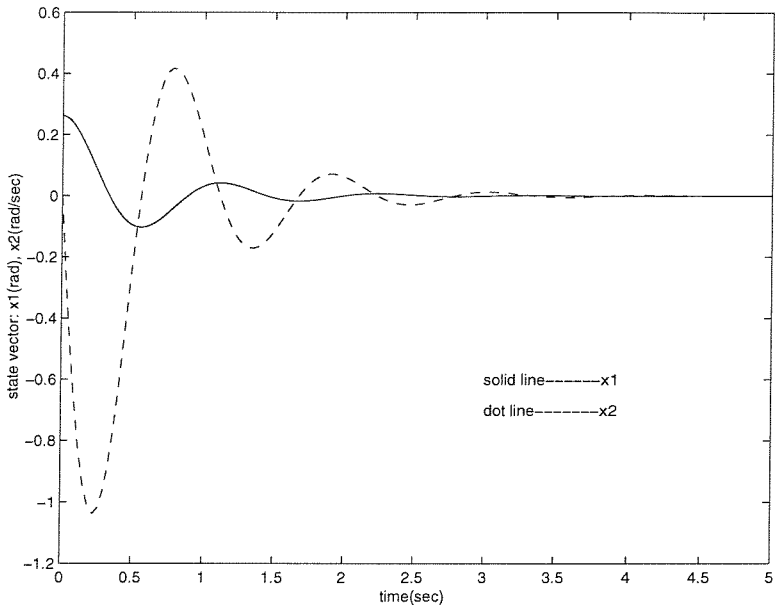

Figure 6.1: State feedback — the angular displacement, x_1, and angular velocity, x_2.

6.4.2 Output feedback control

The techniques derived in section 6.3.1 are used now to design the observers and controllers simultaneously. Suppose that the output matrix in (6.24) is $C = [1 \quad 0]$.

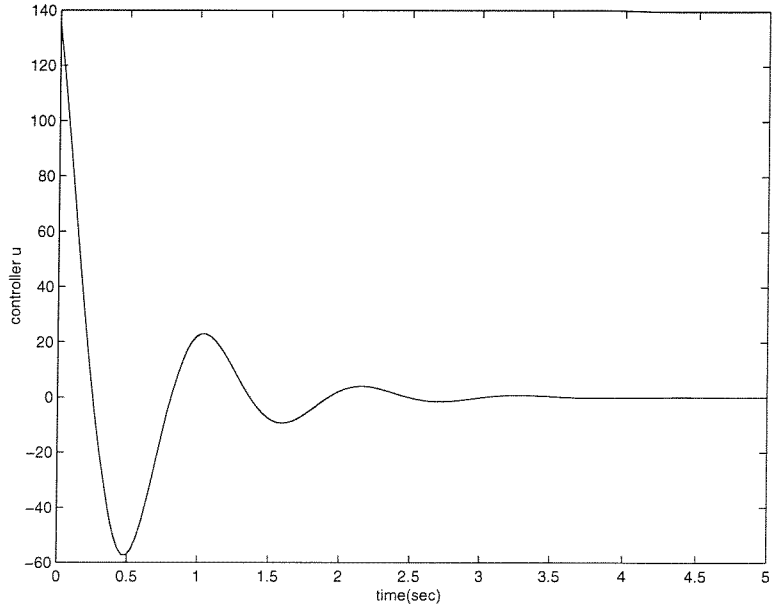

Figure 6.2: State feedback — the controller, thrust, u.

Solving the requisite LMIs (6.41), (6.42) and (6.43) in theorem 6.3, we obtain the following matrices:

$$A_{c1} = \begin{bmatrix} -940.2012 & 238.8003 \\ -1354.7145 & 316.3020 \end{bmatrix}, \quad A_{c2} = \begin{bmatrix} -9.6723 & 6.9901 \\ -5.9111 & 8.7924 \end{bmatrix},$$

$$B_{c1} = \begin{bmatrix} 225.3332 \\ 322.8026 \end{bmatrix}, \quad B_{c2} = \begin{bmatrix} 4.9325 \\ 5.6217 \end{bmatrix},$$

$$K_1 = [1900.2008 \;\; 2483.1123], \quad K_2 = [3.3313 \;\; 7.1212].$$

The corresponding output and input results are depicted in Figure 6.3 and Figure 6.4, respectively.

The figures demonstrate that the performance is satisfactory even though there exists system uncertainty.

This simulation example is well know in the literature and many designs have been done. A typical work is the PDC (Parallel Distributed Compensation) controller presented by Wang, Tanaka and Griffin (1996). But their model does not include modelling mismatch term $\Delta\mathbf{f}(\mathbf{x})$. Kiriakidis (1999) considered $\Delta\mathbf{f}(\mathbf{x})$ in their stabilizing controller design. Both papers only solved the problem of state feedback controller design. Here following our scheme, both state feedback and output feedback controllers can be derived.

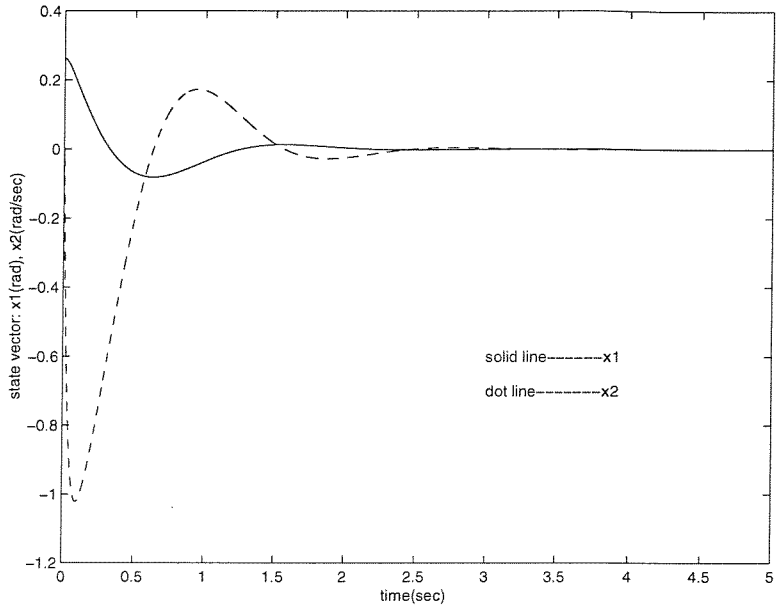

Figure 6.3: Output feedback — the angular displacement, x_1, and angular velocity, x_2.

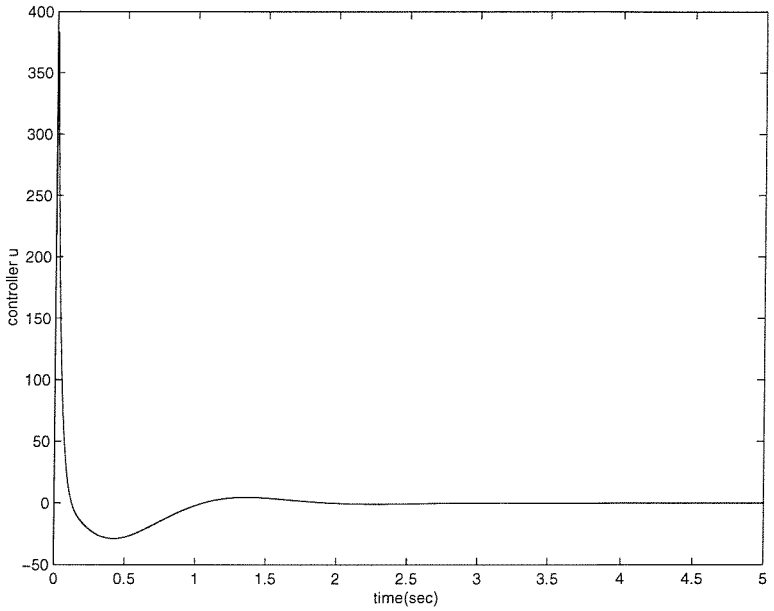

Figure 6.4: Output feedback — the controller, thrust, u.

6.5 Conclusions

In this chapter we have developed a fuzzy model based feedback stabilizing control method that utilize LMIs. The sufficient conditions for stabilization of the feedback systems are given in terms of LMIs. The fuzzy feedback controller can be obtained by solving the LMIs. The technique is robust in the sense that we can handle modelling errors or system perturbations. By applying the piecewise Lyapunov stability conditions which were derived in last chapter, the LMIs involved in the design are less conservative and the usage of commercial LMI solvers provides easy design evaluation.

According to Lemma 6.1, we only need the inequality (6.17) to hold in a local area Z_i. Therefore a further research can be made to further relax the conditions by applying the S-procedure. Anther possible development is to derive a non-linear stabilizing controller by generalizing the above results to parallel distributed compensation (Wang, Tanaka and Griffin, 1996).

Chapter 7

Adaptive Neurofuzzy Control for A Class of State Dependent Nonlinear Processes

In chapter 2, a general MIMO discrete time nonlinear system described by (2.1) was modelled by (2.22), which is rewritten here for easy reference:

$$\begin{aligned}\mathbf{y}(t+1) &= a_1(O_t)\mathbf{y}(t)+\cdots+a_n(O_t)\mathbf{y}(t-n+1)\\ &\quad +b_1(O_t)\mathbf{u}(t-d)+\cdots+b_m(O_t)\mathbf{u}(t-d-m+1)\end{aligned} \tag{7.1}$$

where $\{\mathbf{u}(t),\mathbf{y}(t)\}_{t=1}^{L}$ are measured input/output vector pairs, the integers n and m are known *a priori* or assumed system orders and d is the known time delay of the system, $a_i(O_t)$ and $b_j(O_t)$ $(i=1,2,\cdots,n;\ j=1,2,\cdots,m)$ are *a priori* unknown functions of the measured operating point O_t.

The topic of this chapter concerns the control problems of the system (7.1) (Feng and Harris, 1998). It was remarked in chapter 2 that, if O_t is independent of the system input and output, and (7.1) is SISO, the nonlinear coefficients can be expressed via *associative memory networks* and therefore the system (7.1) can be expressed as

$$y(t+1)=\sum_{k=1}^{n}N_{a_k}(O_t)y(t-k+1)+\sum_{k=1}^{m}N_{b_k}(O_t)u(t-k+1)+\triangle f(O_t), \tag{7.2}$$

where $\triangle\mathbf{f}(O_t)$ is a model mismatching error:

$$\triangle f(O_t)=\sum_{k=1}^{n}(N_{a_k}(O_t)-a_k(O_t))y(t-k+1)+\sum_{k=1}^{m}(N_{b_k}(O_t)-b_k(O_t))u(t-k+1), \tag{7.3}$$

which have already been formulated in (2.25) and (2.26).

If the operating point variations of (7.1) are known then various forms of controller gain scheduling may be deployed. Often these parametric relationships are complex, unknown, and nonlinear but the operating points may be measurable or at least inferred from other sensor outputs. While the general control problem of system (7.1) is considered difficult, researchers have made some progress for a few special cases. For a matching system, by using the parametric (neural net) estimation from equations (2.30) — (2.32), an on-line adaptive controller of the 1-step ahead predictor kind (it has been generalized to d-step ahead for time delay systems) can be easily constructed (Wang, Brown and Harris, 1996) by solving the simplified (for $d = 0$) Diophantine equation

$$1 = \hat{A}(t, q^{-1}) + q^{-1}\hat{G}(t, q^{-1})$$

for a polynomial $\hat{G}(t, q^{-1})$ in the backward shift operator q^{-1}, where

$$\hat{A}(t, q^{-1}) = 1 - \sum_{i=1}^{n} N_{a_i}(O_t) q^{-i}.$$

The 1-step ahead control signal $u(t)$ that ensures $\hat{y}(t+1) = y^*(t+1)$ satisfies

$$y^*(t+1) = \hat{G}(t, q^{-1}) y(t) + \hat{B}(t, q^{-1}) u(t), \tag{7.4}$$

where $y^*(t)$ is the desired output reference signal and $\hat{B}(t, q^{-1}) = \sum_{j=1}^{m} N_{b_j}(O_t) q^{-j+1}$. Stability of the resultant closed loop system is established by the following:

Theorem 7.1 *(Wang, Brown and Harris, 1996) If the system (7.1) is invertibly stable (ie, the bounded desired output $y^*(t)$ can be realised by the bounded control signal), and the basis functions $\phi_k(O_t)$ are uniformly bounded, the resultant matching system is closed-loop stable under the controller derived from (7.4), i.e.*

$$|y(t)| < \infty, \qquad |u(t)| < \infty.$$

For mismatching systems with known common parametric bound $\delta(> 0)$

$$|N_{a_i}(O_t) - a_i(O_t)| \leq \delta,$$
$$|N_{b_j}(O_t) - b_j(O_t)| \leq \delta,$$

for all (i, j) and O_t, such that the total known mismatch model error is

$$|\triangle f(O_t)| < \delta\left(\sum_{k=1}^{n} |y(t-k+1)| + \sum_{k=1}^{m} |u(t-k+1)|\right), \tag{7.5}$$

the above adaptive controller applies, but with an amended parameter estimator, that incorporates the *a priori* determined modeling accuracy (7.5).

The above neural network modeling procedure requires that the input vector O_t is both measurable and bounded. In the more general case when $O_t = \mathbf{x}_t$, the dynamical system (2.1) must be open-loop stable and bounded, somewhat restricting the above methodology. If however a one-to-one mapping can be generated (Wang, Wang, Brown and Harris, 1996) that transforms an unbounded input definition domain into an open set, which is covered by a closed bounded set, then the above modeling and controller design is applicable.

This chapter extends operating point dependent processes form of local modelling and control to the multivariable input multivariable output system of general form (7.1) with known orders, in which $a_i(\cdot)$ and $b_j(\cdot)$ are unknown functions of $\mathbf{x}_t$, the vector of system input and output defined in section 2.1.1. Because the discrete time state space model is going to be used in this chapter, we alter the notations in a slightly different way as:

$$\begin{aligned}
\mathbf{x}(t) &= [\mathbf{x}_1^T(t), \mathbf{x}_2^T(t)]^T \in \Re^{np+mq}, \\
\mathbf{x}_1(t) &= [\mathbf{y}^T(t-n+1), \cdots, \mathbf{y}^T(t)]^T \in \Re^{np}, \\
\mathbf{x}_2(t) &= [\mathbf{u}^T(t-m+1), \cdots, \mathbf{u}^T(t)]^T \in \Re^{mq}, \\
\mathbf{y}(t) &= [y_1(t), \cdots, y_p(t)]^T \in \Re^p, \\
\mathbf{u}(t) &= [u_1(t), \cdots, u_q(t)]^T \in \Re^q.
\end{aligned}$$

In the following it is shown that when the nonlinearities $a_i(\mathbf{x}(t))$ and $b_j(\mathbf{x}(t))$ can be modelled off-line by *artificial neural networks* (ANNs) then a stable decoupling controller can be synthesized. If the coefficients of ANNs are *a priori* unknown, a modified recursive least square(NRLS) algorithm combined with feedback linearising controller is employed to design an adaptive control system. For both cases, the closed loop system stability is analysed in detail and the weight convergence is shown to be guaranteed, for practical implementation the resultant controllers can be realized as conventional controllers or as neurofuzzy controllers (Brown and Harris, 1994).

7.1 Minimum Phase Property

It is well known that certain classes of ANNs can approximate any continuous non-linear function defined on a compact set to an arbitrary accuracy (Brown and Harris, 1994). Therefore, if ANNs are used to approximate the unknown nonlinear functions

in (7.1), estimates for the unknown parameters $a_i(\mathbf{x}(t))$ and $b_j(\mathbf{x}(t))$ can be obtained. In this chapter we use B-spline networks, since they have a direct fuzzy interpretation, and have strong mathematical properties that determine weight stability, convergence and network conditioning (Brown and Harris, 1994).

With O_t being replaced by $\mathbf{x}(t)$ in (7.1) and modifying (2.23) and (2.24) for MIMO systems as

$$N_{a_i}(\mathbf{x}(t)) = \sum_{k=1}^{l} W_{ik}^{a}\phi_k(\mathbf{x}(t)), \quad i = 1,\cdots,n, \tag{7.6}$$

$$N_{b_j}(\mathbf{x}(t)) = \sum_{k=1}^{l} W_{jk}^{b}\phi_k(\mathbf{x}(t)), \quad j = 1,\cdots,m, \tag{7.7}$$

where W_{ik}^{a} is a $p \times p$ matrix, W_{jk}^{b} is a $p \times q$ matrix, and $\phi_k(\mathbf{x}(t))$ is a scalar neurofuzzy basis function, the model representation (7.1) can be rewritten as the following form:

$$\mathbf{y}(t+1) = \sum_{k=1}^{n} N_{a_k}(\mathbf{x}(t))\mathbf{y}(t-k+1) + \sum_{k=1}^{m} N_{b_k}(\mathbf{x}(t))\mathbf{u}(t-k+1) + \triangle\mathbf{f}(\mathbf{x}(t)) \tag{7.8}$$

where

$$\begin{aligned}\triangle\mathbf{f}(x(t)) &= \sum_{k=1}^{n}(N_{a_k}(\mathbf{x}(t)) - a_k(\mathbf{x}(t)))\mathbf{y}(t-k+1) \\ &\quad + \sum_{k=1}^{m}(N_{b_k}(\mathbf{x}(t)) - b_k(\mathbf{x}(t)))\mathbf{u}(t-k+1).\end{aligned}$$

is a vector $\in \Re^p$ and can be represented as $\triangle\mathbf{f}(\mathbf{x}(t)) = (\triangle f_1(\mathbf{x}(t)),\cdots,\triangle f_p(\mathbf{x}(t)))^T$.

The control objective is to track a given bounded reference sequence $\{\mathbf{y}^*(t)\}$ while rejecting the impact of model mismatch error $\triangle\mathbf{f}(\mathbf{x}(t))$. To do this we introduce a *feedback linearising control law* (Sastry and Bodson, 1989) given by

$$\sum_{k=1}^{n} N_{a_k}(\mathbf{x}(t))\mathbf{y}(t-k+1) + \sum_{k=1}^{m} N_{b_k}(\mathbf{x}(t))\mathbf{u}(t-k+1) = \mathbf{r}(t). \tag{7.9}$$

where $\mathbf{r}(t) = (r_1(t),\cdots,r_p(t))^T \in \Re^p$ is an external signal vector to be determined or selected by the designer.

Remark Strictly speaking, the existence of the feedback linearising controller requires the system to be *linearizable* (Sastry and Isidori, 1989) which involves a special matrix of the system being bounded away from singularity. Here we suppose the condition is met, ie., suppose that (7.9) has at least one solution for $\mathbf{u}(t)$ at any time $t \geq 0$. (which is true in particular if the ANNs do not explicitly depend on $\mathbf{u}(t)$ or if they are invertible as a function of $\mathbf{u}(t)$).

Denote the $\mathbf{u}(t)$ implicitly determined by (7.9) as $\mathbf{g}(\mathbf{x}_1(t), \mathbf{x}_2(t-1), \mathbf{r}(t))$. If the basis functions in (7.6) and (7.7) are uniformly bounded, $\mathbf{g}(\mathbf{x}_1(t), \mathbf{x}_2(t-1), \mathbf{r}(t))$ has bounded partial derivatives and so satisfies Lipschitz condition with Lipschitz constant c_g.

Together with (7.9) we can easily obtain the state model for (7.8) as

$$\mathbf{x}_1(t+1) = A_1\mathbf{x}_1(t) + B_1(\mathbf{r}(t) + \triangle\mathbf{f}(\mathbf{x}(t))), \tag{7.10}$$

$$\mathbf{x}_2(t+1) = A_2\mathbf{x}_2(t) + B_2\mathbf{u}(t+1), \tag{7.11}$$

$$\mathbf{y}(t) = H\mathbf{x}_1(t), \tag{7.12}$$

where

$$A_1 = \begin{bmatrix} 0 & I_1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & I_1 & \cdots & 0 & 0 \\ & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & 0 & \cdots & 0 & I_1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ I_1 \end{bmatrix}, \quad H = [0 \cdots 0 I_1],$$

$$A_2 = \begin{bmatrix} 0 & I_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & I_2 & \cdots & 0 & 0 \\ & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & 0 & \cdots & 0 & I_2 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ I_2 \end{bmatrix},$$

where I_1 and I_2 are $p \times p$ and $q \times q$ identity matrices, respectively.

Model (7.10) — (7.12) is in the *normal form*. If it starts from $\mathbf{x}_1(0) = 0$, $\mathbf{r}(t) \equiv 0$ and $\triangle\mathbf{f}(\mathbf{x}(t)) \equiv 0$, then $\mathbf{x}_1(t) \equiv 0$ and the plant output, y, stays at zero. The motion of the resultant system is determined by the dynamics of $\mathbf{x}_2(t)$. We then define the *zero dynamics* of (7.10) — (7.12) as

$$\mathbf{x}_2(t+1) = A_2\mathbf{x}_2(t) + B_2\bar{\mathbf{u}}(t+1), \tag{7.13}$$

where $\bar{\mathbf{u}}(t+1)$ is a function of $\mathbf{x}(t)$ implicitly defined by

$$N_{b_1}(\mathbf{x}(t+1))\bar{\mathbf{u}}(t+1) + N_{b_2}(\mathbf{x}(t+1))\bar{\mathbf{u}}(t) + \cdots + N_{b_m}(\mathbf{x}(t+1))\bar{\mathbf{u}}(t-m+2) = 0.$$

The nonlinear system (7.10) — (7.12) is said to be *globally exponentially minimum phase* if its zero dynamics (7.13) are globally exponentially stable independent of $\mathbf{x}_1(t)$.

It can be shown (Chen and Khalil, 1994; Sastry and Bodson, 1989)) that if (7.13) is exponentially stable there exists a Lyapunov function $V_2(\mathbf{x}_2(t))$ such that

$$c_1\|\mathbf{x}_2(t)\|^2 \leq V_2(\mathbf{x}_2(t)) \leq c_2\|\mathbf{x}_2(t)\|^2, \tag{7.14}$$

$$V_2(\mathbf{x}_2(t+1)) - V_2(\mathbf{x}_2(t)) \leq -\alpha\|\mathbf{x}_2(t)\|^2, \tag{7.15}$$

$$\left\| \frac{\partial V_2(\mathbf{x})}{\partial \mathbf{x}} \right\| \leq L\|\mathbf{x}\| \tag{7.16}$$

hold in some ball $B_u \subset \Re^m$, where c_1, c_2, α and L are constants.

For the purpose of this chapter the norm of a transfer function $H(q^{-1})$ is defined as

$$\|H(q^{-1})\|_\infty = sup_{\omega\in[0,2\pi)}|H(e^{-j\omega})|.$$

7.2 Decoupling control

Let the tracking error be

$$\begin{aligned} \tilde{\mathbf{y}}(t) &= \mathbf{y}(t) - \mathbf{y}^*(t)\ (=[\tilde{y}_1(t),\cdots,\tilde{y}_p(t))^T], \\ \tilde{\mathbf{x}}_1(t) &= [\tilde{\mathbf{y}}(t-n+1),\cdots,\tilde{\mathbf{y}}(t)]^T. \end{aligned}$$

Selecting the external signal

$$\mathbf{r}(t) = \mathbf{y}^*(t+1) - \frac{S(q^{-1})}{T(q^{-1})}\tilde{\mathbf{y}}(t), \tag{7.17}$$

where $\frac{S(q^{-1})}{T(q^{-1})}\tilde{\mathbf{y}}(t)$ is defined as

$$\frac{S(q^{-1})}{T(q^{-1})}\tilde{\mathbf{y}}(t) = \left[\frac{S_1(q^{-1})}{T_1(q^{-1})}\tilde{y}_1(t),\cdots,\frac{S_p(q^{-1})}{T_p(q^{-1})}\tilde{y}_p(t)\right]^T,$$

we obtain from (7.10) that

$$\tilde{\mathbf{x}}_1(t+1) = A_1\tilde{\mathbf{x}}_1(t) + B_1G(q^{-1}) \triangle \mathbf{f}(\mathbf{x}(t)), \tag{7.18}$$

where $G(q^{-1}) \triangle \mathbf{f}(\mathbf{x}(t))$ is defined as

$$G(q^{-1}) \triangle \mathbf{f}(\mathbf{x}(t)) = (G_1(q^{-1}) \triangle f_1(\mathbf{x}(t)),...,G_p(q^{-1}) \triangle f_p(\mathbf{x}(t)))^T,$$

where $G_h(q^{-1}) = \frac{T_h(q^{-1})}{T_h(q^{-1})+q^{-1}S_h(q^{-1})}$ for $h = 1,...,p$ are transfer functions in the delay operator q^{-1}, which can be chosen to be exponentially stable.

A_1 is a stable matrix since all of its eigenvalues are at the origin. Therefore, given any symmetric $Q > 0$, there is a symmetric $P > 0$ such that

$$A_1^T P A_1 - P = -Q. \tag{7.19}$$

Theorem 7.2 *If*

(i) $\mathbf{y}^*(t)$ *is bounded, with bound* $c^* = sup_{t\geq 0}\|\mathbf{y}^*(t)\|$;

(ii) *The transfer functions* $G_h(q^{-1})$ *are proper and exponentially stable, hence there are* $\rho \in (0,1)$ *and* $c_\rho \geq 0$ *such that the impulse response coefficients of* $G_h(q^{-1})$ *are all bounded by* $c_\rho \rho^t$ *for all* $t \geq 0$;

(iii) *(7.10) — (7.12) is globally exponentially minimum phase;*

(iv) *All basis functions in (7.6) and (7.7) are uniformly bounded;*

(v) *There exists a constants* $\delta \in (0,1)$ *such that*

$$\delta < \frac{\|Q\|}{2\|A^TPB\|\left(\frac{c_\rho}{1-\rho}\right) + \|B^TPB\|\left(\frac{c_\rho}{1-\rho}\right)^2}, \tag{7.20}$$

$$\|\triangle \mathbf{f}(\mathbf{x}(t))\| \leq \delta\|\tilde{\mathbf{x}}_1(t)\|, \tag{7.21}$$

then the control law (7.9) and (7.17) results in bounded tracking, that is, $\mathbf{x}(t) \in \Re^{np+mq}$ *is bounded and* $\lim_{t\to\infty}\mathbf{y}(t) = \mathbf{y}^*(t)$.

Proof Choose Lyapunov function candidate for $\tilde{\mathbf{x}}_1(t)$ as

$$V_1(\tilde{\mathbf{x}}_1(t)) = \tilde{\mathbf{x}}_1^T(t)P\tilde{\mathbf{x}}_1(t),$$

where P is determined by (7.19). Then, using (7.20) and (7.21),

$$\begin{aligned}
V_1(\tilde{\mathbf{x}}_1(t+1)) - V_1(\tilde{\mathbf{x}}_1(t)) &= -\tilde{\mathbf{x}}_1^T(t)Q\tilde{\mathbf{x}}_1(t) + 2\tilde{\mathbf{x}}_1^T(t)A_1^TPB_1(G(q^{-1})\triangle\mathbf{f}(\mathbf{x}(t))) \\
&\quad + B_1^TPB_1(G(q^{-1})\triangle\mathbf{f}(\mathbf{x}(t)))^2 \\
&\leq -\tilde{\mathbf{x}}_1^T(t)Q\tilde{\mathbf{x}}_1(t) + 2\|A_1^TPB_1\|\left(\frac{c_\rho}{1-\rho}\right)\delta\|\tilde{\mathbf{x}}_1(t)\|^2 \\
&\quad + \|B_1^TPB_1\|\left(\frac{c_\rho}{1-\rho}\right)^2\delta^2\|\tilde{\mathbf{x}}_1(t)\|^2 \\
&\leq -\tilde{\mathbf{x}}_1^T(t)Q\tilde{\mathbf{x}}_1(t) + \left(2\|A_1^TPB_1\|\frac{c_\rho}{1-\rho}\right. \\
&\quad \left. + \|B_1^TPB_1\|\left(\frac{c_\rho}{1-\rho}\right)^2\right)\delta\|\tilde{\mathbf{x}}_1(t)\|^2 \\
&< -\tilde{\mathbf{x}}_1^T(t)Q\tilde{\mathbf{x}}_1(t) + \|Q\|\|\tilde{\mathbf{x}}_1(t)\|^2 \\
&\leq 0.
\end{aligned}$$

Therefore $V_1(\tilde{\mathbf{x}}_1(t))$ is a lower bounded and decreasing series. Hence

$$\lim_{t\to\infty} \tilde{\mathbf{x}}_1(t) = 0,$$

which means that $\mathbf{x}_1(t)$ is bounded and $\lim_{t\to\infty} \mathbf{y}(t) = \mathbf{y}^*(t)$.

Next, consider the Lyapunov function $V_2(x_2(t))$ in (7.14)—(7.16). Denoting the solution of (7.13) as $\bar{\mathbf{x}}_2(t)$, we have that

$$\begin{aligned}
V_2(\mathbf{x}_2(t+1)) - V_2(\mathbf{x}_2(t)) &= V_2(\bar{\mathbf{x}}_2(t+1)) - V_2(\mathbf{x}_2(t)) \\
&\quad + V_2(\mathbf{x}_2(t+1)) - V_2(\bar{\mathbf{x}}_2(t+1)) \\
&\leq -\alpha\|\mathbf{x}_2(t)\|^2 + \left\|\frac{\partial V_2(\mathbf{x})}{\partial \mathbf{x}}\right\| \|\mathbf{x}_2(t+1) - \bar{\mathbf{x}}_2(t+1)\| \\
&\leq -\alpha\|\mathbf{x}_2(t)\|^2 + L\|\mathbf{x}_2(t)\|\|\mathbf{x}_2(t+1) - \bar{\mathbf{x}}_2(t+1)\| \\
&= -\alpha\|\mathbf{x}_2(t)\|^2 + L\|\mathbf{x}_2(t)\|\|\mathbf{g}(\mathbf{x}_1(t), \mathbf{x}_2(t-1), \mathbf{r}(t)) \\
&\quad - \mathbf{g}(0, \mathbf{x}_2(t-1), 0)\| \\
&\leq -\alpha\|\mathbf{x}_2(t)\|^2 + L\|\mathbf{x}_2(t)\| c_g(\|\mathbf{x}_1(t)\| + \|\mathbf{r}(t)\|) \\
&\leq -\alpha\|\mathbf{x}_2(t)\|^2 + c_3\|\mathbf{x}_2(t)\|,
\end{aligned}$$

where $c_3 = Lc_g sup_{t\geq 0}(\|\mathbf{x}_1(t)\| + \|\mathbf{r}(t)\|)$, which exists since $\mathbf{x}_1(t)$ is bounded. So

$$V_2(\mathbf{x}_2(t+1)) - V_2(\mathbf{x}_2(t)) \leq 0,$$

if $\|\mathbf{x}_2(t)\| \geq \frac{c_3}{\alpha}$.

Using this along with the bounds in (7.14), it is easy to establish that $\mathbf{x}_2(t)$ is bounded, establishing the theorem.

In section 7.4, examples will be given to illustrate the technique developed in this section.

7.3 Adaptive Control

7.3.1 Parameter Estimation

For MIMO systems, we need to extend the definitions of θ and $\Phi(t)$ in (2.28) and (2.29). Redefine $\Phi(t)$ as

$$\Phi(t) = [\phi_1 \mathbf{x}^T(t), \cdots, \phi_l \mathbf{x}^T(t)]^T,$$

and define θ_h as

$$\theta_h = \begin{bmatrix} \theta_{h1} \\ \cdots \\ \theta_{hl} \end{bmatrix}, \qquad \theta_{hi} = \begin{bmatrix} \mathbf{w}_{ni}^{ha} \\ \cdots \\ \mathbf{w}_{1i}^{ha} \\ \mathbf{w}_{mi}^{hb} \\ \cdots \\ \mathbf{w}_{1i}^{hb} \end{bmatrix}, \qquad \begin{pmatrix} h = 1, \cdots, p \\ i = 1, \cdots, l \end{pmatrix},$$

where vector $\mathbf{w}_{ki}^{ha}$ is the transpose of the h-th row of the matrix W_{ki}^{a}, $k = 1, \cdots, n$, and $\mathbf{w}_{ji}^{hb}$ is the transpose of the h-th row of the matrix W_{ji}^{b}, $j = 1, \cdots, m$. Then

$$y_h(t+1) = \Phi^T(t)\theta_h + \triangle f_h(\mathbf{x}(t)). \tag{7.22}$$

according to (7.8). If θ is unknown, the following NRLS algorithm can be used to estimate these parameters.

As suggested in Johansen (1994a), define the normalized signal $n(t)$ as

$$n(t) = \beta n(t-1) + \|\mathbf{x}(t)\| \tag{7.23}$$

where $\beta \in (0,1)$ is a constant and $n(0) > 0$. Denote

$$\begin{aligned} \mathbf{y}_n(t) &= \mathbf{y}(t)/n(t), \\ \Phi_n(t-1) &= \Phi(t-1)/n(t), \\ \triangle_n \mathbf{f}(\mathbf{x}(t)) &= \triangle \mathbf{f}(\mathbf{x}(t))/n(t). \end{aligned}$$

Then the normalized equation

$$y_{nh}(t+1) = \Phi_n^T(t)\theta_h + \triangle_n f_h(\mathbf{x}(t)) \tag{7.24}$$

can be obtained from (7.22). Assume that $\triangle \mathbf{f}(\mathbf{x}(t))$ satisfies

$$\| \triangle \mathbf{f}(\mathbf{x}(t)) \| < \delta_1 \|\mathbf{x}(t)\| \tag{7.25}$$

for some constant δ_1. Then from (7.23) it follows that

$$\| \triangle_n \mathbf{f}(\mathbf{x}(t)) \| < \delta_1. \tag{7.26}$$

Consider the following NRLS algorithm:

$$\hat{\theta}_h(t) = \hat{\theta}_h(t-1) + \frac{\eta(t)P(t-2)\Phi_n(t-1)\varepsilon_n(t)}{1+\Phi_n^T(t-1)P(t-2)\Phi_n(t-1)}, \tag{7.27}$$

$$\varepsilon_{nh}(t) = y_{nh}(t) - \hat{y}_{nh}(t), \tag{7.28}$$

$$\hat{y}_{nh}(t) = \Phi_n^T(t-1)\hat{\theta}_h(t-1), \tag{7.29}$$

$$P(t-1) = \left(I - \frac{\eta(t)P(t-2)\Phi_n(t-1)\Phi_n^T(t-1)}{1+\Phi_n^T(t-1)P(t-2)\Phi_n(t-1)} \right) P(t-2), \tag{7.30}$$

where

$$\eta(t) = \begin{cases} 1 & \text{if } |\varepsilon_{nh}(t)| > \nu(t), \\ 0 & \text{if } |\varepsilon_{nh}(t)| \leq \nu(t), \end{cases} \tag{7.31}$$

and

$$\nu(t) = \frac{\delta_1}{(1+\Phi_n^T(t-1)P(t-2)\Phi_n(t-1))^{1/2}}. \tag{7.32}$$

Theorem 7.3 *If*

(i) *The algorithm (7.27)—(7.32) is applied to the input and output data* $\{\mathbf{u}(t), \mathbf{y}(t)\}$ *of (2.1);*

(ii) *The modeling error is bounded such that* $\| \triangle \mathbf{f}(\mathbf{x}(t))\| < \delta_1 \|\mathbf{x}(t)\|$,

Then, for $h = 1, ..., p$,

$$\begin{aligned} &\lim_{t\to\infty} \quad \frac{\varepsilon_{nh}^2(t)}{1+\Phi_n^T(t-1)P(t-2)\Phi_n(t-1)} = \delta_1^2, \\ &\lim_{t\to\infty} \quad \|\hat{\theta}_h(t) - \hat{\theta}_h(t-k_h)\| = 0, \end{aligned}$$

where k_h *are positive integers.*

Proof See Wang, Brown and Harris (1996).

7.3.2 The Controller

Replace the unknown parameters in (7.9) by $\hat{\theta}_h$, which gives the *certainty equivalence* feedback controller

$$\hat{N}_{a_1}\mathbf{y}(t) + \cdots + \hat{N}_{a_n}\mathbf{y}(t-n+1) + \hat{N}_{b_1}\mathbf{u}(t) + \cdots + \hat{N}_{b_m}\mathbf{u}(t-m+1) = \mathbf{r}(t),$$

where $\mathbf{r}(t)$ determined by (7.17). With this control algorithm, the closed loop is described by

$$\mathbf{y}(t+1) = \mathbf{r}(t) + \sum_{k=1}^{n}(N_{a_k} - \hat{N}_{a_k})\mathbf{y}(t-k+1) + \sum_{k=1}^{m}(N_{b_k} - \hat{N}_{b_k})\mathbf{u}(t-k+1) + \triangle\mathbf{f}(\mathbf{x}(t)),$$

and so

$$\tilde{y}_h(t) = G_h(q^{-1})(-\Phi^T(t-1)\tilde{\theta}_h(t-1) + \triangle f_h(\mathbf{x}(t))). \tag{7.33}$$

The stability of the closed loop adaptive system is given by the following theorem.

Theorem 7.4 *If the assumptions (i)—(iv) of theorem 7.2 hold, and in addition,*

(i) *The modeling error is bounded such that* $\| \triangle \mathbf{f}(\mathbf{x}(t)) \| < \delta_1 \|\mathbf{x}(t)\|$;

(ii) $\left\| \frac{c_4}{1-\beta q^{-1}} \sum_{h=1}^{p} \left(c_5 + \frac{c_6}{1-\sigma q^{-1}} \left(1 + \frac{S_h(q^{-1})}{T_h(q^{-1})} \right) \right) G_h(q^{-1}) \right\|_\infty < 1,$

where c_4, c_5, c_6, *and* σ *are constants (defined in the proof), then for arbitrary initial conditions, all variables in the closed loop are bounded and the tracking error satisfies*

$$\lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t} \|\tilde{\mathbf{y}}(k)\|^2 < \infty.$$

Proof Since (7.13) is globally exponentially stable independent of $\mathbf{x}_1(t)$, there exist constants $c_u > 0$ and $\sigma \in (0,1)$ such that the transition matrix of (7.13), $E(t,t_0)$, satisfies

$$\|E(t,t_0)\| \leq c_u \sigma^{t-t_0}.$$

On the other hand, the solution of (7.11) takes the form

$$\mathbf{x}_2(t) = E(t,0)\mathbf{x}_2(0) + \sum_{k=0}^{t} E(t,k)B(\mathbf{u}(k+1) - \bar{\mathbf{u}}(k+1)).$$

Therefore

$$\begin{aligned}
\|\mathbf{x}_2(t)\| &\leq \|E(t,0)\|\|\mathbf{x}_2(0)\| + \sum_{k=0}^{t} \|E(t,k)\|\|\mathbf{u}(k+1) - \bar{\mathbf{u}}(k+1)\| \\
&\leq \|E(t,0)\|\|\mathbf{x}_2(0)\| + \sum_{k=0}^{t} \|E(t,k)\| c_g (\|\mathbf{x}_1(k)\| + \|\mathbf{r}(k)\|) \\
&\leq c_u \sigma^t \|\mathbf{x}_2(0)\| + \sum_{k=0}^{t} c_u \sigma^{t-k} c_g (\|\mathbf{x}_1(k)\| + \|\mathbf{r}(k)\|) \\
&\leq c_u \sigma^t \|\mathbf{x}_2(0)\| + \frac{c_u c_g}{1-\sigma q^{-1}} (\|\mathbf{x}_1(t)\| + \|\mathbf{r}(t)\|) \\
&\leq c_u \sigma^t \|\mathbf{x}_2(0)\| + \frac{c_u c_g}{1-\sigma q^{-1}} (\|\mathbf{y}(t-n+1)\| + \cdots + \|\mathbf{y}(t)\| + \|\mathbf{r}(t)\|) \\
&= \frac{c_u c_g}{1-\sigma q^{-1}} (\|\mathbf{y}(t-n+1)\| + \cdots + \|\mathbf{y}(t)\| + \|\mathbf{r}(t)\|) + o(\sigma^t),
\end{aligned}$$

where $o(\sigma^t)$ denotes the exponentially decay term. It follows that

$$\|\mathbf{x}_2(t)\| - \sigma\|\mathbf{x}_2(t-1)\| \leq c_u c_g (\|\mathbf{y}(t-n+1)\| + \cdots + \|\mathbf{y}(t)\| + \|\mathbf{r}(t)\|) + o(\sigma^t).$$

From the inequality above we obtain that

$$\begin{aligned}
\|\mathbf{x}_2(t)\| &\leq c_u c_g \sum_{k=1}^{t} \sigma^{t-k}(\|\mathbf{y}(k-n+1)\| + \cdots + \|\mathbf{y}(k)\| + \|\mathbf{r}(k)\|) + o(\sigma^t) \\
&\leq c_u c_g \left(\sum_{l=0}^{n-1} \sum_{k=1}^{t} \sigma^{t-k} \|\mathbf{y}(k-l)\| + \sum_{k=1}^{t} \sigma^{t-k} \|\mathbf{r}(k)\| \right) + o(\sigma^t) \\
&= c_u c_g \left(\sum_{l=0}^{n-1} \sigma^{-l} \left(\sum_{k=l+1}^{t} \sigma^{t-k+l} \|\mathbf{y}(k-l)\| \right) + \sum_{l=1}^{n-1} \sum_{k=1}^{l} \sigma^{t-k} \|\mathbf{y}(k-l)\| \right. \\
&\quad \left. + \sum_{k=1}^{t} \sigma^{t-k} \|\mathbf{r}(k)\| \right) + o(\sigma^t) \\
&\leq c_u c_g \sum_{l=0}^{n-1} \sigma^{-l} \sum_{k=1}^{t} \sigma^{t-k} \|\mathbf{y}(k)\| + c' \sigma^t + c_u c_g \sum_{k=1}^{t} \sigma^{t-k} \|\mathbf{r}(k)\| + o(\sigma^t) \\
&\leq c_u c_g \left(\sum_{l=0}^{n-1} \sigma^{-l} \frac{1}{1-\sigma q^{-1}} \|\mathbf{y}(t)\| + \frac{1}{1-\sigma q^{-1}} \|\mathbf{r}(t)\| \right) + o(\sigma^t) \\
&\leq \frac{c_6}{1-\sigma q^{-1}} \|\mathbf{y}(t)\| + \frac{c_u c_g}{1-\sigma q^{-1}} \|\mathbf{r}(t)\| + o(\sigma^t),
\end{aligned}$$

where $c' = c_u c_g \sum_{l=1}^{n-1} \sum_{k=1}^{l} \sigma^{-k} \|\mathbf{y}(k-l)\|$ and $c_6 = c_u c_g (1 + \sigma^{-1} + \cdots + \sigma^{-n+1})$ are constants. Similarly,

$$\begin{aligned}
\frac{1}{1-\beta q^{-1}} \|\mathbf{x}_1(t)\| &\leq \frac{1}{1-\beta q^{-1}} (1 + \beta^{-1} + \cdots + \beta^{-n+1}) \|\mathbf{y}(t)\| + o(\beta^t) \\
&\leq \frac{c_5}{1-\beta q^{-1}} \|\mathbf{y}(t)\| + o(\beta^t),
\end{aligned}$$

where $c_5 = 1 + \beta^{-1} + \cdots + \beta^{-n+1}$ is a constant.

It can be seen that

$$\begin{aligned}
n(t) &= \frac{1}{1-\beta q^{-1}}\|\mathbf{x}(t)\| \le \frac{1}{1-\beta q^{-1}}(\|\mathbf{x}_1(t)\|+\|\mathbf{x}_2(t)\|) \\
&\le \frac{1}{1-\beta q^{-1}}\left(c_5\|\mathbf{y}(t)\|+\frac{c_6}{1-\sigma q^{-1}}(\|\mathbf{y}(t)\|+\|\mathbf{r}(t)\|)+o(\sigma^t)\right)+o(\beta^t) \\
&\le \frac{1}{1-\beta q^{-1}}\left(c_5\|\mathbf{y}(t)\|+\frac{c_6}{1-\sigma q^{-1}}\|\mathbf{y}(t)\|\right. \\
&\quad \left.+\frac{c_6}{1-\sigma q^{-1}}\left(\|\mathbf{y}^*(t+1)\|+\left\|\frac{S(q^{-1})}{T(q^{-1})}\tilde{\mathbf{y}}(t)\right\|\right)+o(\sigma^t)\right)+o(\beta^t) \\
&\le \frac{1}{1-\beta q^{-1}}\left(c_5+\frac{c_6}{1-\sigma q^{-1}}\right)(\|\tilde{\mathbf{y}}(t)\|+c^*)+\frac{c_6c^*}{(1-\beta q^{-1})(1-\sigma q^{-1})} \\
&\quad +\frac{c_6}{(1-\beta q^{-1})(1-\sigma q^{-1})}\sum_{h=1}^{p}\frac{S_h(q^{-1})}{T_h(q^{-1})}|\tilde{y}_h(t)|+o(\sigma^t)+o(\beta^t) \\
&= \frac{1}{1-\beta q^{-1}}\sum_{h=1}^{p}\left(c_5+\frac{c_6}{1-\sigma q^{-1}}\left(1+\frac{S_h(q^{-1})}{T_h(q^{-1})}\right)\right)|\tilde{y}_h(t)|+c''+o(\sigma^t)+o(\beta^t),
\end{aligned}$$

where $c''=\frac{1}{1-\beta q^{-1}}\left(c_5+\frac{2c_6}{1-\sigma q^{-1}}\right)c^*$ is a constant.

Alternatively, as the basis functions $\phi_k(x(t))$ in (7.6) and (7.7) are uniformly bounded, there exists a constant c_4 such that the $\varepsilon_{nh}(t)$ in (7.28) satisfies (Wang, Brown and Harris, 1996)

$$|\varepsilon_{nh}(t)| < c_4.$$

From (7.24), (7.28), (7.29) and (7.33),

$$\begin{aligned}
\tilde{y}_h(t) &= G_h(q^{-1})(-\Phi^T(t-1)\tilde{\theta}_h(t-1)+\triangle f_h(\mathbf{x}(t))) \\
&= G_h(q^{-1})\varepsilon_{nh}(t)n(t). \qquad (7.34)
\end{aligned}$$

Hence

$$\begin{aligned}
n(t) \le\ & \frac{1}{1-\beta q^{-1}}\sum_{h=1}^{p}\left(c_5+\frac{c_6}{1-\sigma q^{-1}}\left(1+\frac{S_h(q^{-1})}{T_h(q^{-1})}\right)\right)G_h(q^{-1})c_4n(t) \\
& +c''+o(\sigma^t)+o(\beta^t).
\end{aligned}$$

Denote

$$F(q^{-1})=\frac{c_4}{1-\beta q^{-1}}\sum_{h=1}^{p}\left(c_5+\frac{c_6}{1-\sigma q^{-1}}\left(1+\frac{S_h(q^{-1})}{T_h(q^{-1})}\right)\right)G_h(q^{-1}).$$

According to assumption (ii) of the theorem, $\|F(q^{-1})\|_\infty < 1$. Also, according to assumption (ii) of theorem 3, $n(t)$ is a bounded sequence and

$$n(t) \le \frac{c''}{1-\|F(q^{-1})\|_\infty}+o(\sigma^t)+o(\beta^t).$$

So $\|\mathbf{x}(t)\| = (1-\beta q^{-1})n(t)$ is also bounded. Furthermore, from (7.34),

$$\begin{aligned}\|\tilde{\mathbf{y}}(t)\|^2 &= \sum_{h=1}^{p} |G_h(q^{-1})\varepsilon_{nh}(t)n(t)|^2 \\ &= \sum_{h=1}^{p} \frac{c_\rho c''}{(1-\rho)(1-\|F(q^{-1})\|_\infty)}|\varepsilon_{nh}(t)| + o(\sigma^t) + o(\beta^t) \\ &\leq \sum_{h=1}^{p} \frac{c_\rho c'' c_4}{(1-\rho)(1-\|F(q^{-1})\|_\infty)} + o(\sigma^t) + o(\beta^t).\end{aligned}$$

It is trivial to derive from the preceding results that

$$\lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t} \|\tilde{\mathbf{y}}(t)\|^2 < \infty.$$

The theorem has been proved.

7.4 Simulation Examples

Rather than introduce our own examples we demonstrate the efficacy of our approach with the bench-mark example:

$$\begin{aligned}y(t+1) &= \frac{1.5y(t)y(t-1)}{1+y^2(t)+y^2(t-1)} + 0.7sin(0.5(y(t)+ \\ &\quad y(t-1)))cos(0.5(y(t)+y(t-1))) + 1.2u(t). \qquad (7.35)\end{aligned}$$

A simplified version of this process was considered for control design by Sastry and Isidori (1989) with complete plant knowledge, Chen and Khalil (1994) extended this to the case when the gains (1.5, 0.7 and 1.2) were *a priori* unknown via a neural network. In the following examples we consider the same example, but the only *a priori* knowledge utilised is the process model orders, the model structure and nonlinearities being assumed unknown throughout.

7.4.1 Example 1: System with time varying parameters

Suppose that the unknown system (7.35) has in addition some unknown time varying parameters:

$$\begin{aligned}y(t+1) &= \frac{p_1(t)y(t)y(t-1)}{1+y^2(t)+y^2(t-1)} + p_2(t)sin(0.5(y(t) \\ &\quad +y(t-1)))cos(0.5(y(t)+y(t-1))) + p_3(t)u(t), \qquad (7.36)\end{aligned}$$

	$0 \leq \text{rem}(t/100) < 30$	$30 \leq \text{rem}(t/100) < 60$	$60 \leq \text{rem}(t/100) < 100$
$p_1(t)$	1.5	5	10
$p_2(t)$	0.7	0.9	0.4
$p_3(t)$	1.2	2	4

Table 7.1: The time varying parameters

where $p_1(t)$, $p_2(t)$ and $p_3(t)$ are time varying functions listed in Table 7.1 to evaluate the robustness capability of the proposed methodology to parametric changes, where $\text{rem}(t/100)$ denotes the remainer of t divided by 100.

We model the system (7.36) via associative memory networks as

$$\hat{y}(t+1) = N_{a_1}y(t) + N_{a_2}y(t-1) + N_{b_1}u(t). \tag{7.37}$$

where N_{a_1}, N_{a_2} and N_{b_1} are networks each with 49 two-dimensional multivariate B-spline basis functions whose input vector is $(y(t), y(t-1))$. The 7 univariate basis functions for both $y(t)$ and $y(t-1)$ are shown in Figure 7.1.

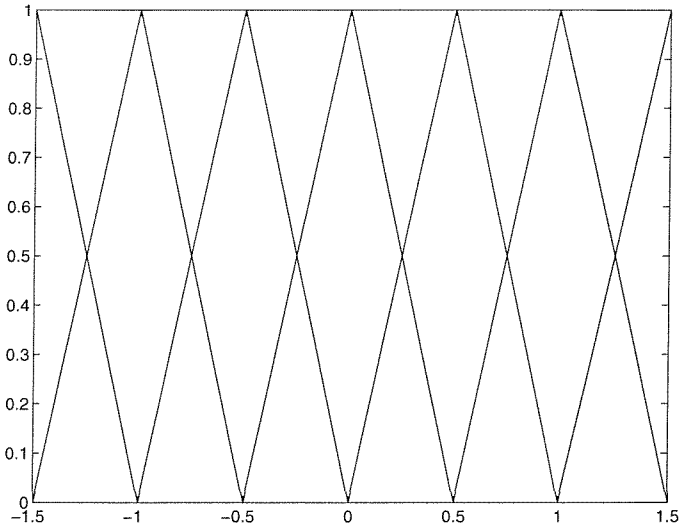

Figure 7.1: Univariate B-spline basis functions for input $y(t)$ or $y(t-1)$

Suppose that we take $y^*(t) \equiv 0$ and $\frac{S(q^{-1})}{T(q^{-1})} \equiv 1$ and so $r(t) = -y(t)$. To perform adaptive control, set $y(0) = 0.4$, $y(1) = 0.2$ and $u(1) = -0.2$. The initial values of weights are chosen randomly in the interval $[-0.1, 0.1]$. The β in (7.23) and the initial matrix $P(0)$ are selected to be 0.2 and $500I$, respectively. The results of the closed loop response are shown in Figure 7.2. The resultant controller is insensitive to temporal parametric change.

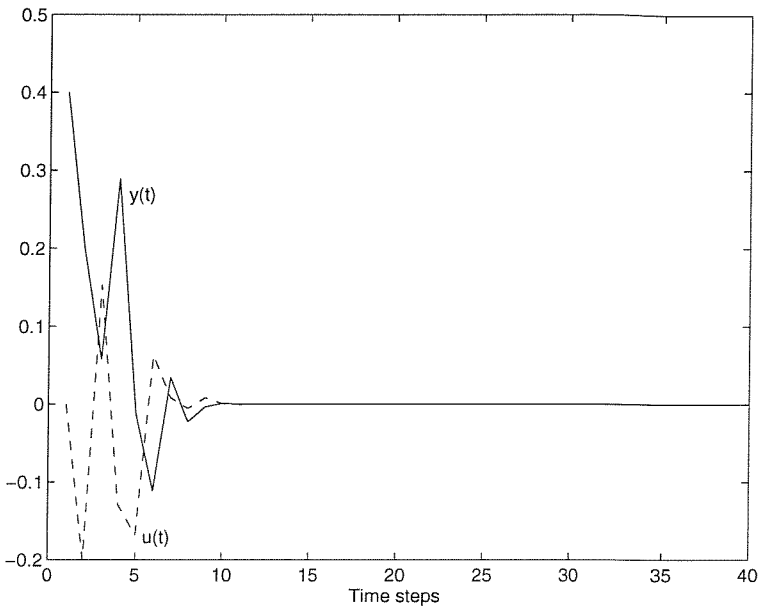

Figure 7.2: The input and output of the adaptive control for Example 7.1

7.4.2 Example 2: Tracking varying reference signal

Suppose that the unknown system is

$$\begin{aligned} y(t+1) &= \frac{1.5y(t)y(t-1)}{1+y^2(t)+y^2(t-1)} + p_2(t)sin(0.5(y(t) \\ &\quad +y(t-1)))cos(0.5(y(t)+y(t-1))) + 1.2u(t), \end{aligned} \tag{7.38}$$

where $p_2(t)$ is same as in Example 1. We now select $y^*(t)$ to be:

$$y^*(t+1) = 0.8y^*(t) - 0.02y^*(t-1) + 0.82z(t),$$

where

$$z(t) = \begin{cases} 0 & \text{if } 0 \leq \text{rem}(t/80) < 40, \\ 4 & \text{if } 40 \leq \text{rem}(t/80) < 80. \end{cases}$$

The rest are taken same as in Section 7.4.1. The results of the closed loop response are shown in Figure 7.3.

7.4.3 Example 3: System with stronger nonlinearity

We now consider a system with a stronger nonlinearity:

$$y(t+1) = \frac{1.5y(t)y(t-1)(1+u(t))}{1+y^2(t)+y^2(t-1)+u^2(t)} \tag{7.39}$$

$$+0.7sin(0.5(y(t)+y(t-1)))cos(0.5(y(t)+y(t-1))). \tag{7.40}$$

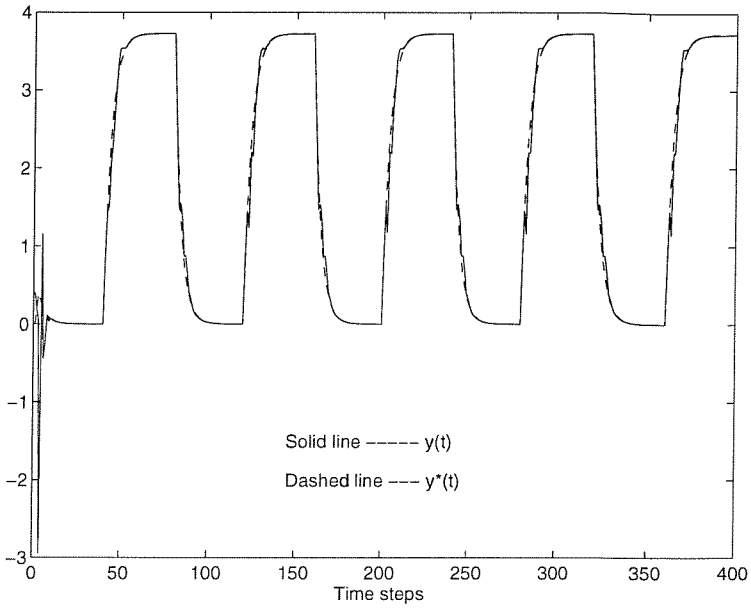

Figure 7.3: The output of the adaptive control for tracking reference signal.

In the previous example, process (7.36) was additive in $u(t)$. In this example, it is not only nonlinear in $u(t)$ but involves an inverse function for which the quasi-linear model may will be inadequate. Despite this the example demonstrates the modeling robustness of this method. Again we use (7.37) to model (7.40). $y^*(t)$ and $\frac{S(q^{-1})}{T(q^{-1})}$ are chosen same as in Section 7.4.1.

7.4.3.1 Decoupled Control

The neural networks in (7.37) are trained using the data $\{u(t), y(t)\}$ of (7.40) for 100 steps. The training algorithm is the NRLS described in Section 7.3.1. During training the $u(t)$ are chosen randomly in $(-1.5, 1.5)$. After the learning process, the control law (7.9) and (7.17) is used to drive the system output from -0.5 to zero. The results are depicted in Figure 7.4.

7.4.3.2 Adaptive Control

For adaptive control design of (7.40), $y(0)$, $y(1)$ and $u(1)$ are set same as in Section 7.4.1, and so as are the initial values of weights, β and the matrix $P(0)$. The results of the closed loop response are shown in Figure 7.5. It can be readily seen that good transient response is achieved.

For completeness the convergence of modeling error during the adaptive control phase is shown in Figure 7.6. The above examples have been subjected to additive

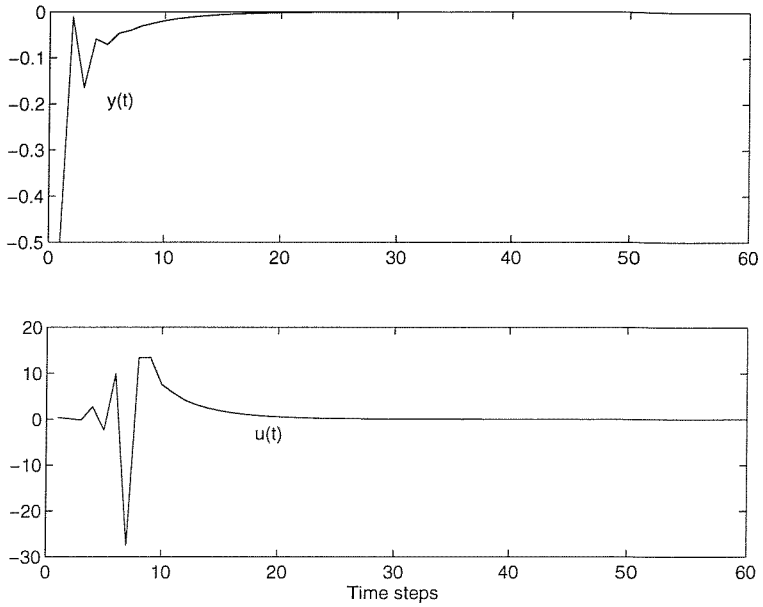

Figure 7.4: The input and output of the decoupling control for (7.40).

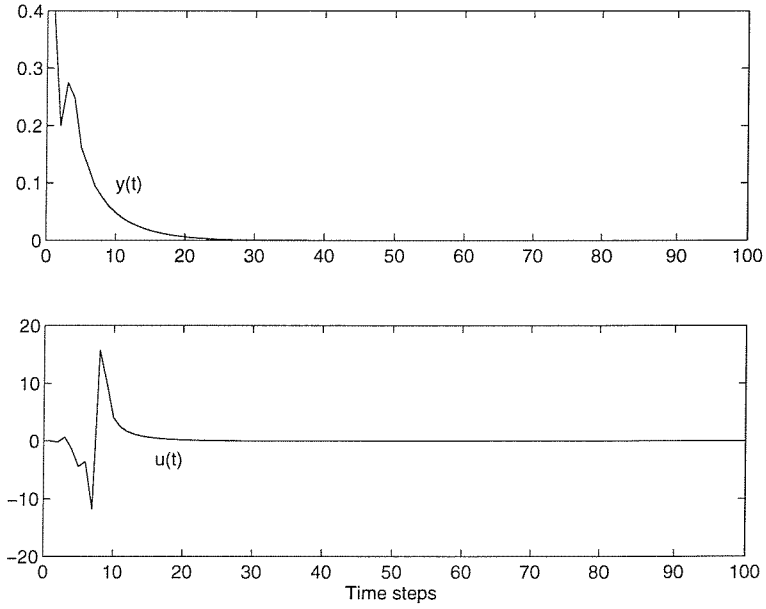

Figure 7.5: The input and output of the adaptive control for (7.40).

noise, the resultant behaviour does not indicate serious deterioration in performance.

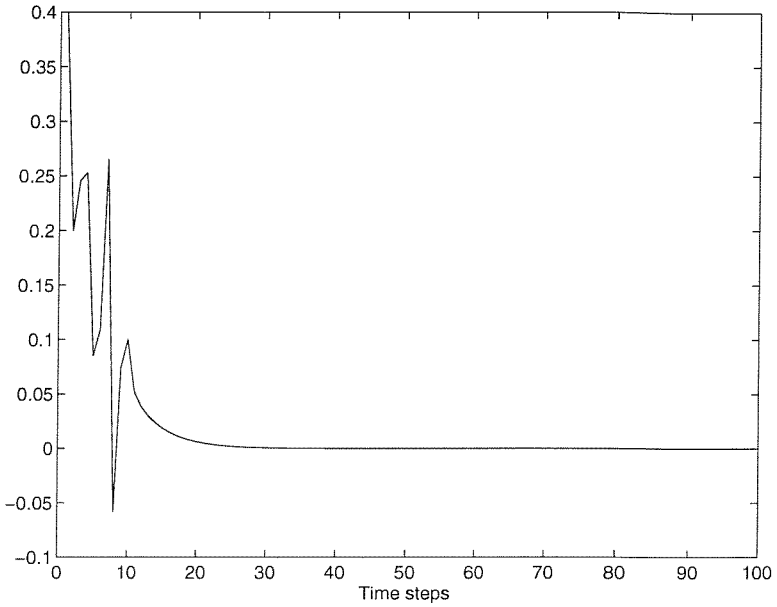

Figure 7.6: The modeling error $y - \hat{y}$.

7.5 Conclusion

In this chapter, a novel neurofuzzy based scheme has been proposed to solve the problem of modeling and control of a class of nonlinear systems in which the observational process parameters are unknown nonlinear functions of the system input and output. Examining the conditions of theorem 7.2 and theorem 7.4 it can be seen that the fundamental assumptions which guarantee the parameter convergence and closed loop stability are minimum phase (assumption (iii) of theorem 7.2) and a limit on the unstructured uncertainty (assumption (v) of theorem 7.2 and assumption (i) of theorem 7.4), which are reasonable from a practical viewpoint. It is interesting to observe from the examples that excellent results can be obtained despite the presence of *a priori* unknown strong nonlinearity in the system and temporal change in process gains. This illustrates the efficacy of combining the neurofuzzy local operating point modeling approach with the classical feedback linearising controller for certain classes of nonlinear system.

Chapter 8

Conclusions

This thesis has dealt with modelling, stability, and control problems of nonlinear systems within local model framework. Both the theoretical aspects and the methodology considerations are emphasized. The results are made easily available through software which implements many of the recent results of LMIs. Several new results were derived while many interesting and important problems remain open research issues. This final chapter gives general conclusions on the research reported in this thesis. More detailed and specific conclusions can be found at the end of the individual chapters. Some open problems and suggestions for further research are also presented.

8.1 Local modelling

Local modelling approaches provide an efficient way for the modelling of complex nonlinear systems. Not only can a multiple local modelling approach be more efficient in capturing the real system dynamics than a single global nonlinear model, but also piecewise locally simpler models (in contrast with the complex global model) are advantageous in that many control and filtering methods are directly applicable to the identified model. For example, piecewise locally linear models enable well-known linear system theory to be directly applicable. In a local model structure, the input space is partitioned into a set of local regions. The local models that operate in these regions are identified separately and the system output is based on a composition of the local models to obtain a good global approximation to the real system.

If a system's behaviour is dependent on some kind of operating point, then a model based on local modelling for the nonlinear system can be obtained in two ways: linearising the system about a set of fixed, known operating points; or partitioning the operating point space into smaller regions and then modelling the system behaviours

locally in the smaller regions. In the latter case, the global model of the system is obtained by combining the local models with some interpolation methods. A typical example of such model structure is fuzzy modelling. Currently there are two categories of methods to partitioning the operating point space, ie., clustering which normally needs to know the number of clusters *a priori*, and operating point space axis orthogonal partitioning which suffers the curse of dimensionality.

In this thesis we attempted to solve the problems discussed above by having developed three kinds of modelling techniques:

1. **A new fuzzy modelling approach** has been developed in chapter 3. This is a technique for constructing data based fuzzy model of a dynamical system by a new method of partitioning the data input space. The method is able to derive a fuzzy model from data automatically and avoids the curse of dimension problem usually associated with fuzzy system.

2. **An optimal piecewise locally linear modelling approach** has been derived in chapter 3. A new algorithm is introduced for the construction of a Delaunay input space partitioned optimal piecewise locally linear models to overcome the COD as well as generate locally linear models directly amenable to linear control and estimation algorithms. The training of the model is configured as a new mixture of experts network with a new fast decision rule derived using convex set theory. A very fast simulated reannealing algorithm is utilised to search a global optimal solution of the Delaunay input space partition.

3. **A class of state dependent NARMAX model structure** has been developed in chapter 7. This is a neurofuzzy based scheme for modeling of a class of nonlinear systems with an ARMA like model (a generalised Takagi-Sugeno fuzzy model), whose parameters are unknown nonlinear functions of the input and output variables or states of the plant. An associative memory network is used to identify each nonlinear function. The main assumptions placed on the system and model for stability are minimum phase and a limit on the modeling mismatch error or uncertainty.

8.2 Local Lyapunov stability

Construction of Lyapunov functions is one of the most fundamental problems of systems theory. The most direct application is of course stability analysis, but analogical problems appear more or less implicitly also in performance analysis, controller

synthesis and system identification. The early results of the Lyapunov stability for neurofuzzy based systems are expressed globally, which is conservative and too strict in practice. The main obstacle to a direct application of the existing techniques is the nontrivial step of finding the appropriate Lyapunov function, which is more like an art than a traceable instrument. On the other hand, the characteristic local property of neurofuzzy modelling prompts the investigation of local Lyapunov stabilities, that is, to develop methods which can search for Lyapunov functions *locally* and combine the local Lyapunov functions to form a global Lyapunov function.

There have been already a couple of methods to deal with this topic. The methods successfully reformulate the problem of searching for Lyapunov functions as a problem of solving a set of LMIs. In this way, the difficulty of finding an appropriate Lyapunov function is overcome by converting the problem to a standard mathematical convex optimizing problem which can be easily implemented by software.

The main drawback that limits the practical use of the above methods is that it may be required to solve a large number of LMIs in the interpolation regions between the system submodels. In addition to the high number of LMIs, the computation complexity and cost also increases dramatically as the input dimensionality increases. This means that the number of parameters involved in the optimization process becomes prohibitively large for large dimensional systems. In this thesis, a new method for the stability analysis of neurofuzzy systems that incorporates the input membership function characteristics is developed in chapter 5. It is shown that, under certain conditions placed on the input membership functions, we need only search for one local Lyapunov function even in the intermodel interpolation region. This both relaxes the stability conditions and reduces the computation load in solving the resultant reduced number of LMIs.

8.3 Controller design for piecewise locally models

Local model based control has emerged as a powerful approach to the controller design of complex non-linear systems. Using the local model structure obtained in this thesis, two important control tasks are addressed in this thesis. One is to stabilize the feedback closed-loop systems via the controller, another one is to design an adaptive controller to achieve tracking performance while guarantees the closed-loop system to be stable.

Often, the first solution depends on finding a common Lyapunov function for the feedback control of nonlinear systems represented by a set of local models. If a nonlinear system is represented by a set of linear TS model, for each local linear

model, a linear feedback controller is designed. The resulting overall controller, which in general is nonlinear, is like the modelling representation, a fuzzy blending of each individual linear controller. The design procedure can finally be expressed as a problem of solving a set of LMIs by using a quasi-linear model structure. More recently, a fuzzy controller design method which attempts to combine individual local linear based solutions to obtain a global solution for the overall design problem has been developed (Cao, Rees and Feng, 1997a; Cao, Rees and Feng, 1997b). The approach is based on the so called fuzzy dynamic model which is an extension of TS model. Instead of searching for a common Lyapunov function for all the subsystems, this algorithm sought for piecewise local quadratic Lyapunov functions. As such it generates a less conservative controller solution than the approaches before. But the resultant controller design is formulated so as to require a set of Riccati equations to be solved with an associated computational overhead.

Following the modelling and stability results obtained, this thesis has formulated and solved the problem of robust stabilization for a broad class of fuzzy systems, in which a class of affine continuous nonlinear dynamical systems are represented by a class of fuzzy models with sector bound modelling errors, for which the state space is partitioned into a set of local linear fuzzy models with bounded additive parametric and modelling errors/disturbances. Both state feedback controller and output feedback controller design methods are derived by seeking a piecewise Lyapunov function for the closed-loop system. The results have two advantages compared with other methods:

1. It is capable of handling modelling error and parametric uncertainty, and

2. by using the stability results derived in this thesis, the design solutions are minimally less conservative and the design process is easy to perform as a problem of solving LMIs.

Concerning the second aspect, a class of nonlinear systems whose parameters are unknown nonlinear functions of the measurable operating point is considered. Before this thesis, a novel approach to the modelling and control for a special case of the system was published (Brown and Harris, 1994; Wang, Brown and Harris, 1996), ie., the SISO case in which the operating point is independent of the system states. But, in practice, this is rarely the case, hence limiting the use of the approach.

This thesis considers the most general case of an MIMO system whose operating point is completely dependent on the system states. It is shown that when the nonlinear coefficients can be modelled off-line by neurofuzzy models then a stable decoupling

controller can be synthesized. If otherwise the coefficients are *a priori* unknown, a modified recursive least square algorithm combined with feedback linearising controller is employed to design an adaptive control system. For both cases, the closed loop system stability is analysed in detail and the weight convergence is shown to be guaranteed, for practical implementation the resultant controllers can be realized as conventional controllers or as neurofuzzy controllers.

8.4 Open problems and ideas for future research

The set of tools and methods for local modelling, control and stability analysis developed in this thesis is not complete nor definitive. Virtually, each presented issue can be investigated in more depth, to yield new insights and extensions. This section discusses some issues based on the results obtained and may be considered as extensions of the material presented in this thesis.

Adaptively partition the operating point space for local modelling

In chapter 3, we have developed a new partitioning approach for fuzzy model construction. The algorithm is able to partition the operating point space recursively in an non-orthogonal manner so that avoids the curse of dimensionality. The framework of the problem formulation was intentionally established in a way that allows the essential aspects of the problem to be examined with ease, ie., in an application-oriented fashion, by the notion of fuzzy quantization. This leads to a natural consideration of extending the approach to *on-line* fuzzy modelling.

In view of this topic, we have noticed that the universal approximation property of fuzzy models has been established in some recent work (Wang, 1994; Wang, 1996), which provides the basis for the on-line fuzzy modelling. The existing schemes normally break the construction problem into two parts. One is the determination of the structure of the fuzzy model, the other is the parameteric estimation. The scheme proposed in Wang (1994), for example, divides the construction into that of initial fuzzy system construction and on-line adaptation. The initialization part determines the structure of the fuzzy system, including the number of fuzzy rules and the locations and shapes of membership functions, while the adaptation part adjusts the parameters of the fuzzy system within a fixed structure. On the other hand, obviously an on-line adaptively fuzzy modelling scheme is more useful in real time control problems.

Examining the fuzzy construction algorithm proposed in chapter 3, we can see that the parameter adaptation is not hard. The key is how to re-partition the operating

point space, ie., choose the new centre point, in each adaptive step. Further research of this problem is both valuable and viable.

Other control schemes for local modelling structure

The locality of control means that the scheme first partitions the operating space of the system into a set of local bounded regions, then the designers select an operating point in each local region and represent the nonlinear system as a simple local system in each region. Finally, they specify or design a local control law in each local region. Because of the complexities of nonlinear systems, to find a set of local control actions is much easier than to find a global control action for the whole systems. Perhaps this is the most important advantage of the local model design methods. In this thesis, only two typical control problems are considered. No doubt there are further classes of control problems concerning local control structure worth further study. The research can progress in at least two ways: one is to apply another kind of control technique to local control structure, ie., adaptive control, predictive control, pole-zero configuration, H^{∞} control, etc. Another is to develop new interpolation schemes such as generalization of the *parallel distributed compensation* (Wang, Tanaka and Griffin, 1996).

Also, according to Lemma 6.1, most inequalities in chapter 6 only need to hold in a local area Z_i. Therefore a further research can be made to relax the conditions by applying the $\mathcal{S}$-procedure.

Integrating stability conditions with modelling schemes

A prospective subject of research is to combine the stability determination with the modelling schemes described in chapter 3, Or develop new local modelling methods with consideration of stability issues.

References

Aiserman, M. A. and F. R. Gantmacher (1965). *Die absolute stabilität von regelsystemen*. R. Oldenburg Verlag.

Aizerman, M. A. and F. R. Gantmacher (1964). *Absolute stability of regulator systems*. Information Systems. San Francisco: Holden-Day.

Akaike, H. (1969). Fitting autoregressive models for prediction. *Ann. Inst. Stat. Math. 21*, 243–247.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Automatic Control 19*(6), 716–723.

Babuska, R. (1996). *Fuzzy Modeling and Identification*. Ph.D. thesis, Delft University of Technology.

Babuska, R. and H. B. Verbruggen (1995a). Identification of composite linear models via fuzzy clustering. In *proceedings European control conference*, Rome, Italy, pp. 1207–1212.

Babuska, R. and H. B. Verbruggen (1995b). New approach to constructing fuzzy relational models from data. In *proceedings* 3rd *European conference on fuzzy and intelligent technologies*, Aachen, Germany, pp. 583–587.

Billings, S. A. and W. S. F. Voon (1987). Piecewise linear identification of non-linear systems. *Int. J. Control 46*, 215–235.

Bossley, K. M. (1997). *Neurofuzzy Modelling Approaches in System Identification*. Ph.D. thesis, University of Southampton.

Bowyer, A. (1981). Computing dirichlet tessellations. *Comp. J. 24*, 162–166.

Box, G. E. P. and G. M. Jenkins (1970). *Time series analysis, forecasting and control*. San Francisco, CA: Holden Day.

Boyd, S., L. E. Ghaoui, E. Feron, and V. Balakrishnan (1994). *Linear matrix inequalities in system and control theory*. SIAM books. Philadelphia: SIAM.

Branicky, M. S. (1995). *Studies in Hybrid Systems: Modeling, Analysis and Control*. Ph.D. thesis, M.I.T., Cambridge, MA.

Brostow, W. and J. P. Dussault (1978). Construction of voronoi polyhedra. *J. Comp. Phys. 29*, 81–92.

Brown, M. and C. J. Harris (1994). *Neurofuzzy adaptive modeling and control*. Prentice Hall International Series in Systems and Control Engineering. London: Prentice Hall.

Cao, S. G., N. W. Rees, and G. Feng (1997a). Analysis and design for a class of complex control systems part i: fuzzy modeling and identification. *Automatica 33*(6), 1017–1028.

Cao, S. G., N. W. Rees, and G. Feng (1997b). Analysis and design for a class of complex control systems part ii: fuzzy controller design. *Automatica 33*(6), 1029–1039.

Chen, F. and H. K. Khalil (1994). Adaptive control of nonlinear systems using neural networks. In C. J. Harris (Ed.), *Advances in Intelligent Control*, pp. 163—181. Taylor & Francis Ltd.

Chen, S. and S. A. Billings (1989). Representation of non-linear systems: The narmax model. *Int. J. Control 49*(3), 1013—1032.

Chen, S. and S. A. Billings (1992). Recursive hybrid algorithm for non-linear system identification using radial basis function networks. *Int. J. Control 55*, 1051–1070.

Chen, S. and B. L. Luk (1999). Adaptive simulated annealing for optimization in nonlinear signal processing applications. *Signal Processing 79*(1), 117–128.

Chillingworth, D. (1976). *Differential topology with a view to applications*. Pitman.

Craven, P. and G. Wahba (1979). Smoothing noisy data with spline functions. estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerical Math 31*, 317–403.

D'Azzo, J. J. and C. H. Houpi (1995). *Linear control system analysis and design: conventional and modern*. New York: McGraw-Hill.

Driankov, D., H. Hellendoorn, and M. Reinfrank (1993). *An introduction to fuzzy control*. Berlin: Springer.

Feng, M. and C. J. Harris (1998). Adaptive neurofuzzy control for a class of state dependent nonlinear processes. *International Journal of System Science 29*, 759–771.

Feng, M. and C. J. Harris (1999a). A new partitioning approach for constructing fuzzy models. In *Proceedings of 3rd World Multiconference on Systemics, Cybernetic and Informatics (SCI'99) and 5th International Conference on Information Systems Analysis and Synthesis (ISAS'99)*, Orlando, Florida, USA. July 31 - August 4.

Feng, M. and C. J. Harris (1999b). Piecewise lyapunov stability conditions of fuzzy systems. Submitted to the IEEE Transactions on SMC.

Feng, M. and C. J. Harris (2000). Feedback stabilization of fuzzy systems via linear matrix inequalities. to appear.

French, M. and E. Rogers (1998). Input/output stability theory for direct neuro-fuzzy controllers. *IEEE Trans. on Fuzzy Systems 6*(3), 331–345.

French, M. and E. Rogers (2000a). Nonlinear iterative learning by an adaptive lyapunov technique. *International Journal of Control (Accepted for publication - special issue on Iterative Learning Control).*

French, M. and E. Rogers (2000b). Stability theory and applications for a class of neuro-fuzzy feedback control schemes. *In : Aracil, J. and Gordillo, F. (eds.) Stability Issues in Fuzzy Control. Springer-Verlag.*

French, M., C. Szepesvari, and E. Rogers (2000a). An asymptotic scaling analysis of lq performance for an approximate adaptive control design. *Submitted to: Mathematics of Control, Signals and Systems.*

French, M., C. Szepesvari, and E. Rogers (2000b). Lq performance bounds for adaptive output feedback controllers for functionally uncertain systems. *Submitted to: Automatica.*

French, M., C. Szepesvari, and E. Rogers (2000c). Uncertainty, performance, and model dependency in approximate adaptive nonlinear control. *IEEE Transactions on Automatic Control 45.*

Gahinet, P., A. Nemirovski, A. Laub, and M. Chilali (1995). *LMI Control Toolbox For use with MATLAB*. The Math Works Inc.

Gustafson, D. and W. C. Kessel (1979). Fuzzy clustering with a fuzzy covariance matrix. In *Proceedings IEEE CDC*, San Diego, CA, USA, pp. 761–766.

Harris, C., M. Brown, K. Bossley, D. Mills, and M. Feng (1996). Advances in neuro-fuzzy algorithms for real-time modelling and control. *J. Engineering Application of AI 9*, 1–16.

Harris, C. J., X. Hong, and M. Feng (1999). Optimal piecewise locally linear modelling. In *Proceedings of SPIE AeroSense'99, Applications and Science of Computational Intelligence II*, Orlando, USA, pp. 486–493.

Harris, C. J., C. G. Moore, and M. Brown (1993). *Intelligent Control: Aspects of fuzzy logic and neural nets*. World Scientific.

Harris, C. J., Z. Q. Wu, and M. Feng (1997). Aspects of the theory and application of intelligent modeling, control and estimation. In *Proc. 2nd Asian Control Conference*, pp. 1—10. Seoul.

Hassibi, A. and S. Boyd (1998). Quadratic stabilization and control of piecewise-linear systems. In *Proceedings of the 1998 American Control Conference*, Philiadelphia, PA, USA, pp. 3659—3664.

Hong, X., C. J. Harris, and P. A. Wilson (1998). Variable selection algorithm for the construction of mimo operating point dependent neurofuzzy networks. Submitted to the IEEE Transactions on Neural Networks.

Hsu, C. S. (1980). A theory to cell-to-cell mapping dynamical systems. *Journal of Applied Mechanics 47*, 931–939.

Hsu, C. S. (1987). *Cell-to-Cell Mapping: A Method of Global Analysis for Nonlinear Systems*. Springer-Verlag.

Huang, P. S., C. J. Harris, and M. S. Nixon (1998). A statistic approach for recognizing humans by gait using spatial-temporal templates. *Proceedings of IEEE International Conference on Image Processing 3*, 178–182.

Ingber, L. (1989). Very fast simulated re-annealing. *Mathematical and Computer Modelling 12*(8), 967–973.

Ingber, L. (1992). Genetic algorithms and very fast simulated reanealling: a comparison. *Mathematical and Computer Modelling 16*, 87–100.

Jain, A. K. and R. C. Dubes (1988). *Algorithms for clustering data.* Englewood Cliffs: Prentice Hall.

Jain, R. (1997). *Neurofuzzy and soft computing*. Prentice Hall.

Jang, J.-S. R., C.-T. Sun, and E. Mizutani (1997). *Neuro-fuzzy and soft computing*. Matlab curriculum series. Upper Saddle River, NJ, USA: Prentice Hall.

Johansen, T. A. (1994a). Fuzzy model based control: Stability, robustness, and performance issues. *IEEE Trans. Fuzzy Systems 2*(3), 221—234.

Johansen, T. A. (1994b). *Operating Regime based Process Modeling and Identification*. Ph.D. thesis, Norwegian Institute of Technology.

Johansen, T. A. and B. A. Foss (1993). Constructing narmax models using armax models. *Int. J. Contr. 58*, 1125—1153.

Johansen, T. A. and B. A. Foss (1995). Identification of nonlinear system structure and parameters using regime decomposition. *Automatica 31*(2), 321–326.

Johansson, M. (1999). *Piecewise Linear Control systems*. Ph.D. thesis, Lund Institute of Technology.

Johansson, M. and A. Rantzer (1998). Computation of piecewise quadratic lyapunov functions for hybrid systems. *IEEE Trans. on Automatic Control 43*(4), 555–559.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica 4*(4), 373–395.

Kavli, T. (1993). Asmod - an algorithm for adaptive spline modelling of observation data. *Int. J. Contr. 58*(4), 947–967.

Kavli, T. (1994). Asmod - an algorithm for adaptive spline modelling of observation data. In H. c j (Ed.), *Advances in intelligent control*, London, pp. Chapter 6. Taylor and Francis.

Kaymak, U. and R. Babuska (1995). Compatible cluster merging for fuzzy modeling. In *Proceedings FUZZ-IEEE/IFES'95*, Yokohama, Japan, pp. 897–904.

Khalil, H. K. (1992). *Nonlinear Systems*. New York: MacMillan.

Kim, E., M. Park, S. Ji, and M. Park (1997). A new approach to fuzzy modeling. *IEEE Trans. on Fuzzy Sys. 5*(3), 328–331.

Kiriakidis, K. (1996). *Analytical methods for stability and synthesis of control systems associated with fuzzy logic*. Ph.D. thesis, Polytechnic University.

Kiriakidis, K. (1999). Non-linear control system design via fuzzy modelling and lmis. *Int. J. Control 72*(7/8), 676–685.

Krishnapuram, R. and C.-P. Freg (1992). Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern recognition 25*(4), 385–400.

Kuo, B. (1980). *Digital control systems*. Holt-Saunders.

Lewis, R. (1990). *Practical digital image processing*. Ellis Horwood series in digital and signal processing. New York: Ellis Horwood Ltd.

Ljung, L. (1987). *System Identification: Theory for the Use*. Englewood Cliffs, NJ.: Prentice-Hall, Inc.

Lo, S. H. (1989). Delaunay triangulation of nonconvex plan domains. *Int. J. Numerical Methods in Engineering 28*, 2695–2707.

Lyapunov, A. M. (1892). The general problem of the stability of motion (in russian). *Collected Works II. Mathematical Society*.

Lyapunov, A. M. (1992). The general problem of the stability of motion (translated into english by a. t. fuller). *Int. J. Control 55*, 531–773. Also published as a book by Taylor & Francis, London.

Nelles, O., S. Sinsel, and R. Isermann (1996). Local basis function networks for identification of a turbocharger. In *UKACC International Conference on CONTROL '96*.

Nesterov, Y. and A. Nemirovsky (1988). A general approach to polynomial-time algorithms design for convex programming. In *Technical report*, Centr. Econ. & Math. Inst., USSR Acad. Sci., Moscow.

Nesterov, Y. and A. Nemirovsky (1994). *Interior-point polynomial methods in convex programming*, Volume 13 of *Studies in Applied Mathematics. SIAM*. Philadelphia, PA.

Ollero, A. and F. Cuesta (1998). *Proceedings of the 1st FAMIMO Workshop on Local Modeling and Local Control.* Escuela Superior de Ingenieros, Dept. Ingenieria de Sistemas y Automatica, Seville, Spain.

Park, J. and I. W. Sandberg (1990). Universal approximation using radial-basis-function networks. *Neural Comput. 3*, 246–257.

Pettersson, S. and B. Lennartson (1997a). Controller design of hybrid systems. In O. Maler (Ed.), *Lecture notes in computer science*, Volume 1201, pp. 240–254. Springer.

Pettersson, S. and B. Lennartson (1997b). Stability analysis of nonlinear and hybrid systems using discontinuous lyapunov functions. *Workshop on Multiple Model Approaches to Modeling and Control*, 29–30.

Polycarpou, M. M. and P. A. Ioannou (1992). Learning and convergence analysis of neural-type structured networks. *IEEE Trans. Neural Networks 3*, 39—50.

Priestley, M. B. (1988). *Non-linear and non-stationary time series analysis.* London: Academic.

Rosen, B. E. (1997). Rotationally parameterized very fast simulated reannealing. *IEEE Trans. Neural Networks.*

Sastry, S. and M. Bodson (1989). *Adaptive control: Stability, convergence, and robustness.* Prentice-Hall Inc.

Sastry, S. and A. Isidori (1989). Adaptive control of linearizable systems. *IEEE Trans. Automatic Contr. 34*, 1123—1131.

Sibson, R. (1978). Locally equiangular triangulations. *Comp. J. 21*, 243—245.

Slotine, J.-J. E. and W. Li (1991). *Applied nonlinear control.* Prentice-Hall.

Smith, R. M. and T. A. Johnansen (1997). *Multiple Model Approaches to Modelling and Control.* Taylor and Francis.

Smith, S. M., B. R. Nokleby, and D. J. Comer (1994). A computational approach to fuzzy logic controller design and analysis using cell state space methods. In Kandel and Langholz (Eds.), *Fuzzy Control Systems*, pp. 398–427. Boca-Raton, Fl.:CRC Press.

Stoer, J. and C. Witzgall (1970). *Convexity and Optimization in Finite Dimensions I.* Springer-Verlag.

Sugeno, M. and G. T. Kang (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems 28*, 15–33.

Sugeno, M. and K. Tanaka (1991). successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems 42*, 315–334.

Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. System, Man and Cybernetics 15*(1), 116–132.

Tan, S. and Y. Yu (1996). Adaptive fuzzy modeling of nonlinear dynamical systems. *Automatica 32*(4), 637–643.

Tanaka, H., S. Uejima, and K. Asai (1982). Linear regression analysis with fuzzy model. *IEEE Trans. System, Man, and Cybernetics 12*(6), 903–907.

Tanaka, K., T. Ikeda, and H. O. Wang (1996). Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: Quadratic stability, h control theory and linear matrix inequalities. *IEEE Trans. Fuzzy Systems 4*(1), 1–13.

Tanaka, K. and M. Sugeno (1992). Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems 45*(2), 135–156.

Tong, R. M. (1979). The construction and evaluation of fuzzy models. In M. M. G. et. al. (Ed.), *Advances in Fuzzy Sets Theory And Its Applications*, pp. 559–576. North-Holland.

Wang, H., M. Brown, and C. J. Harris (1996). Modeling and control of nonlinear, operating point dependent systems via associative memory network. *J. Dynamic and Control 6*(2), 199–218.

Wang, H., C. P. Liu, C. J. Harris, and M. Brown (1995). *Advanced Adaptive Control.* London: J. Wiley.

Wang, H., Y. Q. Liu, and D. H. You (1991). Application of a nonlinear self-tuning controller for regulating the speed of a hydraulic turbine. *Trans. ASME. J. Dyn. Sys. Meas. and Control 113*, 541—544.

Wang, H., A. P. Wang, M. Brown, and C. J. Harris (1996). One-to-one mapping and its application to neural networks based control system design. *Int. J. System Science 27*(2), 161–170.

Wang, H. O., K. Tanaka, and M. F. Griffin (1996). An approach to fuzzy control of non-linear systems: Stability and design issues. *IEEE Trans. on Fuzzy Systems 4*, 14–23.

Wang, L. X. (1994). *Adaptive fuzzy systems and control: design and stability analysis*. Englewood Cliffs, NJ: Prentice-Hall.

Wang, L. X. (1996). *Adaptive fuzzy systems and control*. Englewood Cliffs, NJ: Prentice-Hall.

Wang, L. X. and J. M. Mendel (1992). Fuzzy basis functions, universal approximators and orthogonal least-squares learning. *IEEE Trans. on neural networks 3*, 807–814.

Wu, Z. Q. and C. J. Harris (1998). A mixture expert network for piecewise linear modelling. Available at http://www.isis.ecs.soton.ac.uk, to be published.

Xie, X. and R. Evans (1984). Discrete-time adaptive control for deterministic time-varying systems. *Automatica 20*, 309–319.

Yakubovich, V. A. (1962). The solution of certain matrix inequalities in automatic control theory. *Soviet Math. Dokl. 3*, 620–623.

Yakubovich, V. A. (1964). Solution of certain matrix inequalities encountered in nonlinear control theory. *Soviet Math. Dokl. 5*, 652–656.

Yakubovich, V. A. (1967). The method of matrix inequalities in the stability theory of nonlinear control systems, i, ii, iii. *Automation and Remote Control 25-26*(4), 905–917,577–592,753–763.

Yi, S. Y. and M. J. Chung (1993). Identification of fuzzy relational model and its application to control. *Fuzzy Sets and Systems 59*, 25–33.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. System, Man, and Cybernetics 1*, 28–44.