

BLACK BOX SYSTEM IDENTIFICATION OF A  
NONLINEAR SYSTEM FOR THE DESIGN OF MODEL  
BASED CONTROL BY MULTIMODELS

by

SEOK-BEOM LEE

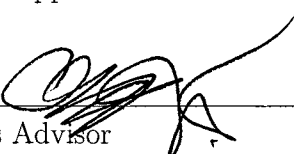
Bachelor of Engineering  
Kon-kuk university  
Seoul, Korea  
1995

Master of Science  
Oklahoma State University  
Stillwater, Oklahoma  
1997

Submitted to the Faculty of the  
Graduate College of the  
Oklahoma State University  
in partial fulfillment of  
the requirements for  
the Degree of  
DOCTOR OF PHILOSOPHY  
May, 2002

BLACK BOX SYSTEM IDENTIFICATION OF A  
NONLINEAR SYSTEM FOR THE DESIGN OF MODEL  
BASED CONTROL BY MULTIMODELS

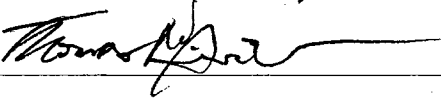
Thesis Approved:

  
\_\_\_\_\_

Thesis Advisor

  
\_\_\_\_\_

  
\_\_\_\_\_

  
\_\_\_\_\_

  
\_\_\_\_\_

Dean of the Graduate College

## **Acknowledgments**

I want to express special thank to Dr. Gary Yen for his generous guidance and patience as well as financial support. I also thank all the professors in my committee for sharing their time and knowledge for the completion of this dissertation. I owe a lot to the professors who taught me the subjects of control and systems, especially Dr. Hagan for introducing me the subject of system identification.

Most of all, my greatest owe is to my parents for their love and trust. I also appreciate the friendship of the people who I met in Stillwater.

# Table of Contents

Chapter	Page
<b>1 Introduction</b>	<b>1</b>
1.1 Overview of system identification . . . . .	2
1.2 Overview of multimodel system identification . . . . .	7
1.3 Motivation of multimodels identification for control . . . . .	12
1.4 Problem formulation and report overview . . . . .	15
1.4.1 Problem formulation for Part I . . . . .	16
1.4.2 Problem formulation for Part II . . . . .	17
1.5 Contributions . . . . .	20
<b>I Multimodels for the identification of linear time varying systems</b>	<b>22</b>
<b>2 Multiple Laguerre models for the identification of linear time varying systems</b>	<b>23</b>
2.1 Introduction . . . . .	23
2.2 Problem formulation . . . . .	25
2.3 Laguerre functions and Laguerre model . . . . .	27
2.4 Heuristic adaptive multiple Laguerre models . . . . .	34
2.4.1 Optimal Laguerre pole estimation . . . . .	36
2.4.2 Heuristic implementation by recursive LS with multiple Laguerre models . . . . .	37
2.5 Adaptive multiple Laguerre models using recursive orthogonal LS with Laguerre poles in irregular grid . . . . .	39
2.5.1 Multiple Laguerre models using irregular grid . . . . .	39
2.5.2 Review of orthogonal least squares methods . . . . .	42
2.5.3 Adaptive Laguerre model selection using orthogonal least squares methods . . . . .	45
2.6 Example . . . . .	49
2.7 Discussion . . . . .	49

<b>II</b>	<b>Study on the local model for multimodel-based nonlinear system identification</b>	<b>52</b>
<b>3</b>	<b>On the local interpretation of Takagi-Sugeno fuzzy models from a dynamical systems point of view</b>	<b>53</b>
3.1	Introduction . . . . .	53
3.2	The relation between TS fuzzy models and dynamical systems . . . .	58
3.3	Example . . . . .	68
3.4	Discussion . . . . .	71
<b>4</b>	<b>On the local identification of a Takagi-Sugeno fuzzy model in state space</b>	<b>74</b>
4.1	Introduction . . . . .	74
4.2	Constant-affine state space model for control . . . . .	75
4.3	CASS model structure for identification . . . . .	79
4.4	Significance of unidentifiability of CASS model . . . . .	90
4.4.1	Optimization method . . . . .	90
4.4.2	Input-output based method . . . . .	96
4.4.3	Bias estimation by augmented states . . . . .	97
4.4.4	Bias estimation by robust observer . . . . .	98
4.5	Discussion . . . . .	112
<b>5</b>	<b>Identification of a CASS model by nonlinear state realization</b>	<b>113</b>
5.1	Introduction . . . . .	113
5.2	Data based state realization of linear system . . . . .	114
5.3	State realization of nonlinear recursive i/o difference equations . . . .	118
5.4	Identification of recursive i/o model for state realization . . . . .	124
5.5	CASS model identification using the state realization . . . . .	126
5.6	Examples . . . . .	128
5.7	Discussion . . . . .	133
<b>6</b>	<b>Conclusion and Future study</b>	<b>134</b>
6.1	Conclusions . . . . .	134
6.2	Further works . . . . .	135
	<b>Bibliography</b>	<b>147</b>
	<b>Appendix</b>	<b>147</b>
<b>A</b>	<b>Overview of unconstrained nonlinear least squares</b>	<b>148</b>
<b>B</b>	<b>Discrete time sliding mode controller</b>	<b>151</b>

# List of Figures

<b>Figure</b>		<b>Page</b>
2.1	Effects of Laguerre pole and order on the accuracy of the Laguerre model	27
2.2	Irregular grid for multiple Laguerre models . . . . .	41
2.3	Adaptive multiple Laguerre model for rapidly switching system identification . . . . .	50
3.1	Error of TS fuzzy model . . . . .	72
5.1	Error between CASS and nonlinear models . . . . .	129
5.2	Training results of nonlinear regression model . . . . .	130
5.3	Validation results of nonlinear regression model . . . . .	131
5.4	States of CASS vs. nonlinear system . . . . .	132

# Chapter 1

## Introduction

The importance of mathematical models in science and engineering cannot be over-phrased. Mathematical models are essential for the representation and generalization of physical phenomena and systems. In practice, mathematical models can be used for simulation and prediction as well as system design and analysis.

On the other hand, obtaining a mathematical model is quite a demanding job that may require empirical testing, mathematical intuition and physical insights, coupled with well-developed as well as understood laws of physics. In engineering applications, the development of a “perfect” model is prohibitive because of the complexity of the systems, limitations in cost and time, noise in measurements and disturbances or immeasurable variables. Therefore, models for engineering applications are often compromised between the practical constraints and the accuracy. *Mechanistic modelling*, which is a conventional approach, is mainly based on expert knowledge and application of the understood physics of a system, and often results in unsolvable complex partial differential equations. On the contrary, *system identification* is to develop models by fitting data into model structure that is inferred from physical intuition, data analysis or for specific application needs. Bear this in mind, system identification is an attractive alternative to the conventional mechanistic modelling.

However, direct application of identified models to model based technologies can be troublesome. Since model structures and parameters are inferred and/or esti-

mated from finite noisy data in system identification, the model is bound to contain high degree of modelling errors, especially for nonlinear systems. Also, since system identification and model based technologies such as model based control have been evolved in parallel, the models in these two disciplines do not necessarily compatible. For example, model based technologies are mostly developed for state space models, in which mechanistic models are derived, while identification models are usually developed in i/o forms.

This work is motivated to narrow the gap between model based technologies and system identifications for nonlinear systems, by extending relatively well developed linear system identification methods into nonlinear systems by multiple models approach. Linear system identification models also have better connection with model based technologies.

The rest of this chapter is organized as follows. In Section 1.1, we overview system identification, emphasizing the differences between linear and nonlinear system identification and the difficulties of application of global nonlinear models to model based technologies. In Section 1.2, an overview of multiple models identification is presented. In Section 1.3, we illustrate the motivations of multimodels based system identification for control, followed by formal problem formulation in Section 1.4. In Section 1.5, the contributions from this study is summarized.

## **1.1 Overview of system identification**

System identification is a subject, theoretical as well as empirical, to build mathematical models of dynamical systems, based on observed data from the systems. The general procedure of system identification consists of: (I) experiment design and data collection, (II) model set or model structure selection, (III) model parameter estimation and (IV) model determination and validation. The most difficult step in system identification is procedure (II). In the choice of model structure, system iden-



tification can be categorized into: (i) gray/black box, (ii) continuous/discrete and (iii) linear/nonlinear.

The categorization of (i) is according to a prior knowledge of model structures and the physical significance of model parameters. In gray box identification, the model structure is obtained from physical intuitions and application of physical laws (Chappell and Godfrey, 1990; Chappell et al., 1999) or to satisfy known properties of the dynamic system such as Lyapunov exponents or equilibrium points (Aguirre, 2000). In contrast to gray box identification, in black box identification, model structures are selected to represent the observations in an efficient way, that is, with least model complexity because of bias and variance tradeoffs (Juditsky et al., 1995). Since gray box identification is system dependent and requires a fair amount of knowledge, *we focus on black box identification* in this study.

The categorization of (ii) does not necessarily imply the use of continuous time data for identification. Rather, the distinction depends on the original model forms. Indeed, discrete time data outweigh continuous time data in their advantages with the modern advancement of digital data acquisition, storage and computation technologies. The need of continuous time system identification is from the fact that a great deal of mechanistic models is derived in continuous time with physically significant but unknown parameters. One intuitive method to estimate the parameters for continuous time models is either to obtain approximate discrete time model by applying numerical integration formula or to numerically differentiate the data (Zhang and Rymer, 1997). In (Johansson et al., 1999), continuous time model is transformed into pseudo-discrete time form by using an operator. However, since we are interested in black box identification, *we only focus on discrete time models*.

The categorization of (iii) is according to model structures and the subsequent differences in estimation and analysis methods. The conventional generic transfer

function model for a given linear system is represented as (Ljung, 1999):

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \quad (1.1)$$

where  $q^{-1}$  is a backward shift operator such that  $q^{-1}u(t) = u(t-1)$ ,  $e(t)$  is white noise and  $A(q), B(q), C(q), D(q), F(q)$  are polynomials of  $q^{-1}$ . This transfer function model has close relation with following model structures:

1. input-output or nonparametric model:

$$\begin{aligned} y(t) &= G(q, \theta)u(t) + H(q, \theta)e(t) \\ &= \sum_{\tau=0}^{\infty} g(\tau)u(t-\tau) + v(t) \end{aligned}$$

2. regression model:

$$y(t) = \varphi(t, \theta)^T \theta$$

3. state space model:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + w_x(t) \\ y(t) &= Cx(t) + Du(t) + w_y(t) \end{aligned}$$

In input-output model,  $g(\tau)$  is an impulse response of the linear system,  $v(t)$  is additive disturbance and usually represented as  $v(t) = \sum_{\tau=0}^{\infty} h(\tau)e(t-\tau)$  where  $e(t)$  is independent random variables with zero mean. Estimation of parameters of the input-output model can be done by truncating the infinite sum to finite terms, which then is equivalent to FIR model identification. This estimation can be achieved efficiently and is robust against noise, however, requires many parameters especially for the identification of lightly-damped systems.

Compared to input-output model, regression model is much more efficient model structure and is closely related to statistical estimation and model building. The

regression model can be also named as a recursive input-output model since the regression vector  $\varphi(t)$  usually consists of past inputs and outputs such as  $\varphi(t) = [y(t-1), \dots, y(t-n_y), u(t-n_k), \dots, u(t-n_u)]^T$ . The regression vector may also contain parameters since regressor of linear models may not be represented as linear regression form even if the model is linear.

State space model is the most closely related to physical systems since many mechanistic models are derived in state space form.  $w_x(t)$  and  $w_y(t)$  can be viewed as the representations of disturbance and measurement noise, respectively. In general,  $w_x(t)$  and  $w_y(t)$  are assumed to be sequences of independent random variables with zero mean and constant covariance. For colored disturbance/noise case, the disturbances/noise can be represented as additional state space model excited by white noise and augmented to the original state space model.

The model structures for nonlinear system identification can be represented in similar ways but in more generalized forms. A nonlinear i/o model structure is Volterra series expansion

$$\begin{aligned}
 y(t) = & \sum_{k=0}^{\infty} a_1(k)u(t-k) + \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} a_2(k_1, k_2)u(t-k_1)u(t-k_2) \\
 & + \dots + \sum_{k_1=0}^{\infty} \dots \sum_{k_n=0}^{\infty} a_n(k_1, \dots, k_n) \prod_{i=1}^n u(t-k_i) + \dots
 \end{aligned}$$

Apparently, the application of Volterra models is very difficult in practice since these models require excessive number of parameters even if the symmetry of the kernels is utilized (Unbehauen, 1996).

Nonlinear state space system can be represented as

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

where  $f$  and  $g$  are general functions. The most popular model structure for the representation of nonlinear state space system is Dynamic Recurrent Neural Networks

(DRNN) (Jin and Gupta, 1995) and is represented as

$$x(t+1) = -\alpha x(t) + h(A, x(t); B, u(t)) \quad (1.2)$$

$$y(t) = Cx(t) \quad (1.3)$$

where  $h(\cdot)$  can be one of the following forms

$$h(t) = A\sigma(x(t)) + Bu(t) \quad (1.4)$$

$$= \sigma(Ax(t) + Bu(t)) \quad (1.5)$$

$$= \sigma(Ax(t)) + Bu(t). \quad (1.6)$$

Here,  $\sigma(\cdot)$  is a differentiable sigmoid function. In (Jin and Gupta, 1995), discrete time DRNN is shown to be able to approximate a discrete-time state-space trajectory uniformly. Funahashi and Nakamura (1993) show that continuous time DRNN can approximate continuous autonomous system. All these proofs are based on the approximation property of feedforward networks using sigmoidal functions. Hence, the convergence of DRNN to state space trajectory is only asymptotic, therefore, the dimension of state space of the recurrent networks is much higher than the dimension of state space of the original systems.

The equivalent form of recursive i/o in nonlinear system identification is

$$y(t) = f(\varphi(t); \theta)$$

where  $f(\cdot)$  is a function mapping regression space  $\varphi(t) \in \Phi$  to output space  $y(t) \in Y$ . To represent dynamics, the regression usually consists of past inputs and outputs, e.g.  $\varphi(t) = [y(t-1), \dots, y(t-n_y), u(t), u(t-1), \dots, u(t-n_u)]^T$ . This form of nonlinear system model is most popular especially because of neural networks and radial basis functions in recent years.

In a strict sense, it is best to infer model structures from the analysis of the measured data using methods summarized in (Haber, 1985; Haber and Unbehauen,

1990). However, identification of nonlinear model structures from data is not trivial, which may be reflected from the popularity of neural networks. Hence, major efforts in nonlinear system identification is currently concentrated on utilizing general model structures having universal function approximation capability. By utilizing this type of model structures, we can reduce the design of nonlinearities of the models into the determination of model size. The problem of this approach is in the asymptotic approximate nature of the models, hence, usually requires very large structure of the model. This makes it difficult to derive inference from the model, to assess the model quality and to select the proper size of the model.

## 1.2 Overview of multimodel system identification

In Section 1.1, we pointed out some of the problems of global nonlinear system identification such as large model structure and high dimension of parameters, which make identification process inefficient and the analysis of the model difficult. Our major motivation of using multiple models is to overcome these limitations of global nonlinear system identification.

Even though many literatures denote multiple models as *divide-and-conquer* approach, there is no strict definition of multiple models. Rather, any system described by more than one model can be denoted as multiple models. There are several reasons of using more than one models for system representation as listed in the following: (i) unevenly distributed data, (ii) identification of switching/hybrid/time-varying systems, (iii) incorporation of a prior knowledge or mechanistic models, (iv) numerical constraints or limitations of model structure and (v) alternative to global system identification.

The case (i) is quite common in practice, especially for nonlinear dynamic systems unless the data are collected in closed loop, since nonlinear dynamic systems can have several stable and unstable regions in the region of interest. It is intuitive that

we cannot develop reliable models for the regions with insufficient data. Fitting a global model for this data set can degrade the overall accuracy of the model. In this sense, using multiple models for case (ii) is natural (Andersson, 1985; Skeppstedt et al., 1992). The case (iii) is the major motivation of the so called *operating regime* based identification and control method (Johansen, 1994). The transparency from the simplicity of local models enables the realization of hybrid models supporting mechanistic models as well as empirical models. The case (iv) is abundant even in linear system identification. For example, for numerical reasons, it may be difficult to adequately describe more than two to three decades of the frequency range within one model (Ljung, 1999, See Chap. 16). The case (v) is probably the major reason to consider multiple model approach, as is considered in fuzzy identification (Jang et al., 1997).

System identification methods by multimodels generally involve the following procedures: (a) partition global data or define locality criterion, (b) identification of local models and (c) coordination of local models for a given operating conditions. These procedures can be approached from deterministic or statistic ways.

Statistical approach to multimodels identification is basically to place models with some hypothesis, modify the parameters of the models iteratively and generate the probability of each hypothesis, conditioned on the residue and a given probability of transition. In (Jordan and Jacobs, 1994), a hierarchical structure is proposed, which is composed of expert networks and gating networks. Expert networks are corresponding to local models, while gating networks are equivalent to coordination rules in multimodel frameworks. Expectation-Maximization (EM) algorithm is used to adjust the parameters of the architecture. In (Skeppstedt et al., 1992), a sequential multimodel identification algorithm is presented. The algorithm consists of three main parts: an estimation algorithm to update parameters by multiple Kalman filters, a classification algorithm based on Bayes' decision rule and a procedure for building

the set of partial models to describe the process.

Deterministic approach to (a) can be divided into unsupervised and supervised partitioning. Unsupervised methods are based on clustering algorithms such as SOM or fuzzy clustering. SOM performs unsupervised clustering of input data to the network (Hagan et al., 1996). For system identification purpose, the input to the SOM network can be selected as  $\varphi(t)$ , the regression vector, by assuming that the unknown system is smooth, that is, the outputs are close as long as regression vectors are close (Principe et al., 1998). A better approach is to use an output augmented vector,  $[\varphi(t)^T, y(t)]^T$  (Ge et al., 1999). SOM using this augmented vector as an input to SOM, combined with some visualization methods such as Umatrix, can be used to identify discontinuity nonlinearities such as backlash and friction (Witkowski et al., 1997). C-means clustering method is another unsupervised clustering algorithm to minimize the cost function of

$$J = \sum_{i=1}^c \sum_{x(t) \in \Omega^i} \|x(t) - \omega^i\|^2 \quad (1.7)$$

where  $\Omega^i$  is defined as

$$\Omega^i = \{x(t) \mid \|x(t) - \omega^i\| < \|x(t) - \omega^j\|, \text{ for all } j \in \{1, \dots, c\}, j \neq i\}. \quad (1.8)$$

Here  $x(t)$  is the feature vector, which is composed of  $\varphi(t)$  or  $[\varphi(t)^T, y(t)]^T$  similarly with SOM based clustering. C-mean clustering is an iterative method utilizing the necessary condition of optimality condition. Since it utilizes necessary condition, there is no guarantee that the iterative procedure converges to the optimal (Jang et al., 1997).

Fuzzy clustering is considered to work better than hard clustering such as SOM or c-means methods because of better statistical properties. Fuzzy c-mean clustering method is to minimize the cost function of

$$J = \sum_{i=1}^c \sum_{t=1}^N \mu_i(t)^m d_i(t)^2 \quad (1.9)$$

where  $d_i(t)^2 = \|x(t) - \omega^i\|_2^2$  and  $m > 1$  is a parameter that controls fuzziness of the clusters. The higher values of  $m$  is, the more clusters overlap.  $\mu_i(t)$  is the soft boundary function between 0 and 1 which satisfies the property of

$$\sum_{i=1}^c \mu_i(t) = 1, \text{ for all } t. \quad (1.10)$$

Since this clustering is based on unweighted 2-norm, the clusters are limited to be hyper-balls. Improved performance can be achieved by using weighted 2-norm, that is,

$$d_i(t)^2 = (x(t) - \omega^i)^T M_i (x(t) - \omega^i). \quad (1.11)$$

In (Gustafson and Kessel, 1979), necessary conditions for optimal  $M_i$ ,  $\mu_i(t)$ , and  $\omega^i$  are derived. In clustering based partition methods, the number of clusters must be specified beforehand. For fixed local model structures, the number of clusters for system representation is a function of the type of nonlinearity of the unknown system. Without a prior knowledge of the nonlinearity of the systems, the number of clusters is usually set to be large and is reduced iteratively combined with local model identification, by using validity measure (Rojas et al., 2000), cluster merging (Murray-Smith and Johansen, 1997, see chap. 5), or considering the significance of individual fuzzy clusters (Setnes, 2000).

One possible problem of the clustering based partition is that the partition is only based on the density of the training data, while local model structure is not considered. An alternative partitioning method is to recursively partition the training data or the region of interest, in order to minimize a certain cost function that is a function of the model error. CART (Classification And Regression Tree) is a binary tree method to minimize a cost function, e.g.

$$J = \sum_{i=1}^m \sum_{(y(t), u(t)) \in \Omega_i} (y(t) - \hat{y}^i(t))^2 \quad (1.12)$$

where  $y(t)$  is the measured output and  $\hat{y}^i(t)$  is  $i$ th local model output.  $t = 0, \dots, N-1$  where  $N$  is the number of training samples.  $\Omega_i$  is the set of  $(u(t), y(t))$ , which is



decomposed to fit  $i$ th local model.  $m$  is the number of local model that has to be estimated by recursively partitioning the region of interest and re-estimation of local models using the re-partitioned data. Even though CART try to decompose the global data set into proper local data set to find the optimal local model, the decomposition of continuum space with binary trees will lead to infinite number of combinatorial problem (Jang et al., 1997). Some heuristics have been suggested as a remedy in (Johansen and Foss, 1995).

We can see from the overview of partitioning algorithms for multimodels identification that partitioning of data without a prior knowledge of the system is not trivial. The tree algorithms, however, do not necessarily lead to isolate good local dynamics which is proper for the local model structure since the partitioning rules are heuristic in nature to avoid the infinite combinatorial problem. Clustering algorithms have advantages over tree algorithms when it is used iteratively with actual identification of model.

The procedure (b) has not been treated seriously, especially for off-line identification. Since multimodel approach is usually taken to take advantage of linear or linear-in-the-parameters model structures, linear recursive i/o model or affine recursive i/o model is used in general. Once the global data are properly assigned, local models can be estimated by well-known linear least squares. However, local model identification can be tricky with overlapping data over multiple local regions, and can fail to represent the local dynamics (Murray-Smith and Johansen, 1997, Chap. 7). Indeed, identification of multimodels has the nature of a multi-objective problem to tradeoff between global and local model accuracy (Yen et al., 1998). Another interesting method is to find the parameters of a local model such that

$$V_i^T([\varphi(t)^T, y(t)]^T - \omega^i) = 0 \quad (1.13)$$

where  $V_i$  is the eigenvector of the smallest eigenvalue of the  $i$ th cluster's covariance

matrix and  $\omega^i$  is the center of the cluster (Murray-Smith and Johansen, 1997, Chap. 2). The idea of this method is that clustering the data, spread around the regression surface, results in flat hyperellipsoids that can be seen locally as hyperplanes.

The procedure (c) can be achieved in many different ways. The most intuitive method would be switching to the local model that minimizes certain criterion, which is a function of residue, given an i/o data or operating conditions (Narendra and Balakrishnan, 1997). For the models based on SOM clustering, the weights of SOM network can be used to achieve soft (Ge et al., 1999) as well as hard switching (Principe et al., 1998). Coordination of local models is natural in fuzzy identification and modelling, however, improper selection of interpolation rules can have adverse effects (Murray-Smith and Johansen, 1997, Chap. 8).

As is clear from the overview of multimodel identification, the decomposition or localization process is quite heuristic in the sense that the process is based on the distribution of data or heuristic decomposition instead of local dynamics that the local models may represent. Also, local model structures have never been taken seriously, but some conjectures from linear system identification or static function approximation are used to infer local model structures.

### **1.3 Motivation of multimodels identification for control**

The goals of automatic control can be categorized into three: (i) automation, (ii) disturbance/fault rejection and (iii) performance/stability enhancement. Even though a great majority of automatic control is nothing but algorithms implemented in computer codes, the development of control algorithms is highly demanding and requires extensive knowledge about the physics of the system to be controlled as well as mathematics for the controller design, since these algorithms interact with real physical systems.

The conventional and most popular control methodology is so called *model based control* (Slotine and Li, 1991; Khalil, 1995; Ogata, 1995). In this methodology, physical systems are thoroughly analyzed and controllers are developed to have at least theoretically guaranteed stability and performance, based on mathematical models of the systems. This type of controllers even provides robustness against uncertainties from modelling errors and unknown disturbances by explicitly considering the degree of model error or disturbances (Zhou and Doyle, 1998; Green and Limebeer, 1995). The major limitation of this method is on the heavy dependence on mathematical models of the system, while the development of mathematical models for control is not trivial and often poses the bottle neck in the control system development.

The difficulties of model development for model based control are attributed by several reasons as follows: (i) obtaining mathematical models for a system requires thorough knowledge about the system, (ii) mathematical models from physics are usually complex high order partial differential equation, (iii) empirical models in various science and engineering disciplines are usually focused only on steady-state phenomena and (iv) mathematical model for control must be simple enough for mathematical analysis and controller design while it must contain the essential dynamics of the system.

The difficulty of model development for model based control spurred different control system development methodology. This methodology is named as *input-output based* or *intelligent* or *model-free* control (Narendra and Parthasarathy, 1990; Hunt et al., 1992; Narendra, 1996). The basic idea of intelligent control is to learn controller or system model by adjusting parameters of a selected model structure by solving minimization of certain error criterion. The major approaches can be divided into direct and indirect control. In indirect control, the generation of controller is by utilizing the model of the system. The motivation of indirect control is that the popular backpropagation algorithm cannot be applied to directly identify the

controller, since the closed loop error is not directly linked to the controller. By identifying models for the system, the controller can be trained by backpropagation method (Narendra and Parthasarathy, 1990). On the other hand, the direct control is to generate controller without the help of mathematical models of the system. One example of direct control is by extending the model based adaptive control to radial basis function networks (Sanner and Slotine, 1992).

Even though intelligent control has been flourishing in literatures, the practical implication of this methodology is questionable. The problem is contributed from the difficulty of identifying general nonlinear system with large and highly nonlinear models, lack of theoretical tools in the analysis of the system for essential properties such as stability, and the robustness against model uncertainty or disturbances. There have been intensive efforts to overcome the criticism of lack of systematic theories, however, most of works are mere applications of nonlinear model based control to the identified models. To see that nonlinear model based control is still an evolving discipline, the extension to learning systems must be considered immature.

On the other hand, the difficulty of nonlinear model based control has been well taken and some alternative approaches have been emerging such as hybrid control or gain scheduling (Antsaklis, 1998; Apkarian et al., 1995). The conventional gain scheduling is motivated to extend linear control into nonlinear systems by divide-and-conquer, that is, closely related to multiple model approach. The problem of the conventional gain scheduling is in the heuristic connection of the linearly designed controllers and lack of analysis of the overall performance of the closed-loop system. In recent years, different approaches have been proposed in the context of gain-scheduling design (Apkarian and Adams, 1998). These approaches reformulate a nonlinear system into linear time or linear parameter varying systems, instead of considering a family of linearized models. In contrast to the conventional gain scheduling, these approaches are based on sound theoretical analyses. However, these approaches

involve conservativeness and restrictions on the systems.

Hybrid system is the recently flourishing subject to handle heterogeneous systems including switching system, digital control systems and discrete-event systems with automaton (Antsaklis, 1998). Hence, the difficulty of nonlinear system is not the sole motivation of the hybrid system. However, the sound theories of supervisory control or switching systems can be considered as an effort to extend the conventional control to cover wider and sometimes discontinuous nonlinear regions.

Intelligent control is not an exception to this new trend. Similar approaches have also been proposed in intelligent control literatures utilizing the famous LMI approach (Kiriakidis et al., 1998; Kiriakidis, 1998).

From the overview of the current control literatures, it is obvious that there are great efforts to develop theoretically sound control methodologies for systems represented by multiple models. Hence, the motivation of our work is to provide useful modelling tools for these new trend of control by system identification. Multiple model identification algorithms, in combination with control methods, are expected to be strong and handy tools for practical as well theoretical purposes for control system development.

## **1.4 Problem formulation and report overview**

The main body of this dissertation are divided into two parts. Part I is for the identification of linear time varying systems using multimodels and Part II is for the local model identification of general nonlinear systems in state space for multimodels. The order of the presentation of the two seemingly different topics are in accordance with how our understanding of multimodels identification for control has been evolved through this project. Part I was originally motivated from the so called multiple model approaches to control, which is usually used to refer some heuristic divide-and-conquer type control (Karimi and Landau, 2000; Murray-Smith and Johansen, 1997).

Based on the conjecture that some nonlinear system can be adequately represented by adapting linear models efficiently, we try to extend the results in linear system identification for control to linear time varying systems by multimodels. However, because of the heuristic of multiple model control and the difficulty of designing controller with adaptive models, we turn our attention to off-line system identification for the design of model based controllers in Part II. In this section, we present the problem formulations of each part.

### 1.4.1 Problem formulation for Part I

Consider a system which can be represented by a set of linear systems as

$$y(t) = G_\omega(q)u(t) \quad (1.14)$$

where the system set  $\Omega$  is defined as

$$\Omega = \{G_\omega(q) \mid \omega : \text{condition of system validity}\}. \quad (1.15)$$

The conditions of validity are for switching the operating system to the most representative system in the system set. We may easily find physical plants that can be modelled by this linear model set, e.g. flexible transmission with varying loading conditions (Karimi and Landau, 2000), set point control of chemical process in (Nystöm et al., 1999), etc.

Suppose that the system is unknown. Our ultimate objective as control engineers would be to identify the system such that we can design a robust controller for the uncertain system. The robustness of controller is essential in order to increase the confidence of the control system as well as to reduce the tuning procedure. In linear robust control, model uncertainty description is represented as

$$y(t) = (G(q) + \Delta G(q))u(t) \quad (1.16)$$

where  $\Delta G(q)$  is the model uncertainty that is unknown but with known bounds. While the uncertainty bounds can be obtained from physical knowledge about the

system and the model, estimation of the model uncertainty is not trivial for uncertain systems.

In system identification, the most popular approach is to adopt basis functions approach that can be represented as

$$y(t) = \left( \sum_{k=1}^{\infty} \theta_k B_k(q) \right) u(t). \quad (1.17)$$

Because of the linear-in-the-parameters structure of the model, analysis of the estimated model is facilitated, which is essential for reliable model development. Also, because of the completeness of basis functions in linear space, model uncertain bounds can be obtained from the difference between an infinite series representing the ‘true’ system and a truncated series representing the nominal model (Hakvoort, 1994; Tøffner-Clausen, 1996).

Motivated from system identification for control and literatures in multiple model control, we try to extend the basis function based system identification to the system described by (1.14). We consider Laguerre basis functions, one of the simplest but useful basis functions, for the purpose. Therefore, the problem that is dealt with in Part I is to develop a system identification methodology by multiple Laguerre basis functions for the identification of system that can be described by (1.14).

### 1.4.2 Problem formulation for Part II

Inspired by system theories, the most general description of a system can be given as (Liu and Moog, 1994)

$$F(y(t), \dot{y}(t), \dots, y^{(n)}(t), u(t), \dot{u}(t), \dots, u^{(s)}(t)) = 0 \quad (1.18)$$

where  $u(t)$  is the input and  $y(t)$  is the output. For ease of exposition, we only consider Single Input Single Output (SISO) system in this report. The objective of system identification is generally to develop a model, given i/o data,  $\{u(kT), y(kT)\}, k = 0, \dots, N - 1$ , where  $N$  is the number of data and considered to be finite for a practical

reason.  $T$  is the sampling period. Since only i/o discrete-time data is available, the natural as well as most common system identification model structure is given as the regression form

$$y(t) = g(y^{t-1}, u^{t-1}). \quad (1.19)$$

The use of i/o history for regression vector is inferred from observations of dynamical systems, which is interpreted as systems with memory. It is obvious that finding a link between (1.18) and (1.19) is very unclear.

To find a link between the continuous time system and discrete-time data, we may discretize (1.18). However, there is no discretization method available for high order differential equations to the author's knowledge. Numerical integration method may be used for the purpose but may require extremely high sampling rate for reasonably accurate model. High sampling is problematic for system identification since sampled data system at high sampling rate *appears* to be a slowly changing system, e.g. poles converging to 1 on the unit circle for linear systems.

Instead, we obtain state realization of (1.18) as

$$\dot{x}(t) = f_c(x(t), u(t)) \quad (1.20)$$

$$y(t) = h_c(x(t), u(t)). \quad (1.21)$$

with  $c$  implies continuous time system. It is known that state realization of the above form does not necessarily exist for (1.18) (Liu and Moog, 1994). We restrict the system set to have the conventional state realization of (1.20)-(1.21). Also, we suppose that the state realization is not redundant, that is, minimal. The definition of minimal realization is given in later chapters. (1.20)-(1.21) is the most common model form considered in control theories. As a matter of fact, input-affine state space form that is usually considered for differential-geometry setting (Isidori, 1995)



as

$$\dot{x}(t) = f_1(x(t)) + f_2(x(t))u(t) \quad (1.22)$$

$$y(t) = h_1(x(t)) \quad (1.23)$$

is only a special form of (1.20)-(1.21). Hence, (1.20)-(1.21) can be considered as quite a general model to represent nonlinear dynamics.

Now, the discretization of continuous time high order system is much facilitated. By proper sampling, it is reasonable to consider the resulting discrete time state space system as the data-generating system. The resulting discrete-time system is given as

$$x((k+1)T) = f(x(kT), u(kT)) \quad (1.24)$$

$$y(kT) = h(x(kT), u(kT)) \quad (1.25)$$

In this report,  $kT$  is replaced with  $t$ , which is the commonly used notation in system identification literatures.

Another motivation of considering (1.24)-(1.25) as the data generating system is that this conventional state space form is the model form that has been extensively studied in control theories. Since we are interested in making clear connection with control, it is reasonable to develop models closely related to this model form.

In general, system identification for control has the following meanings: (i) model must be simple enough, (ii) model has to contain enough information and dynamics, and, furthermore, (iii) model is equipped with model uncertainty description for robust control. However, it is not trivial to identify (1.24)-(1.25) with reasonable simplicity as well as accuracy. Global nonlinear model structures, to cope with the complexity of global nonlinear dynamics, has to be a complex nonlinear function containing large number of parameters. The reliable estimation of the large parameter vector also requires huge training data set that requires excessive load of computation and storage.

In order to facilitate the problems of global nonlinear model structures, we investigate the multimodel based nonlinear system identification, with local model structures in state space. Hence, the objectives of the study are: (i) to investigate the feasibility of identifying (1.24)-(1.25) with multimodels approach, and (ii) to develop a multimodel identification methodology for control.

## 1.5 Contributions

We summarize the contribution from this dissertation in this section.

**Chapter 2: Multiple Laguerre models for the identification of linear time varying systems** Laguerre basis functions and Laguerre models are reviewed. The proof for the optimality of Laguerre basis functions is given and the incorrect result in literatures is corrected. A formula for optimal pole estimation by numerical search is given, and heuristic idea of realizing adaptive multiple Laguerre models is presented. Irregular grid generation for multiple Laguerre models is proposed by utilizing a property of Laguerre basis functions. Switching/parameter estimation algorithm using recursive orthogonal least squares is developed.

**Chapter 3: On the local interpretation of Takagi-Sugeno fuzzy models from a dynamical systems view.** The problem of Takagi-Sugeno (TS) fuzzy models for dynamical interpretation is identified. Analysis is given to reveal the misleading information of the TS fuzzy models and the problem of TS fuzzy model based control is identified.

**Chapter 4: Identification of a deterministic constant-affine state space model with known initial conditions.** Constant-affine state space (CASS) model is proposed as a proper model for model based fuzzy control. Clear notion of the equivalence of CASS models is proposed. Analysis shows that CASS model is not

structurally identifiable. This identifiability property of CASS model is interpreted in several ways by considering the conventional as well as non-conventional estimation methods. In the course, two different discrete-time sliding controllers are derived in constructive ways.

**Chapter 5: Identification of a CASS model by nonlinear state realization.**

A data based state realization method is proposed for linear systems for the identification of state space models. This idea is attempted to be extended into nonlinear systems for the identification of CASS model. State realization and recursive i/o model structure for state realization are reviewed, emphasized on data based approaches.

## **Part I**

# **Multimodels for the identification of linear time varying systems**

# Chapter 2

## Multiple Laguerre models for the identification of linear time varying systems

### 2.1 Introduction

Laguerre basis functions are one of the simplest but useful orthonormal basis functions to approximate linear dynamical systems and have nice recursive structures. In the last decade, there have been considerable interests and developments in using Laguerre basis functions, as well as more general orthonormal functions, in approximation and identification of linear systems, especially for control.

These interests in orthonormal basis functions are due to the nice properties of orthonormal basis functions such as: (i) Orthonormal basis functions are complete in some Banach space of time-invariant systems, that is, the subspace spanned by orthonormal basis functions asymptotically converges to the Banach space. Hence, orthonormal basis functions can be used to obtain quantitative model errors in terms of the difference between an infinite series representing the ‘true’ system and a truncated series representing nominal model (Hakvoort, 1994; Tøffner-Clausen, 1996). (ii) Orthonormal basis functions are proven to be optimal basis functions in *n-width* sense for linear systems belonging to  $H_2$  or  $H_\infty$ , that is, have optimal convergence rate and requires least number of parameters for approximation of the system (Pinkus, 1985;

Mäkilä and Partington, 1993). (iii) Models incorporating orthonormal basis functions are linear-in-the-parameters once basis functions are determined, and statistical properties of the models can be easily analyzed from linear estimation theory. (iv) A prior knowledge of the system can be easily incorporated in terms of dominant poles or time constant since the poles of basis functions are closely related to the system poles (Mäkilä and Partington, 1993; Wahlberg, 1994). The incorporation of a prior knowledge in terms of basis function poles makes the basis function approach much more efficient than FIR models. The incorporation of a prior knowledge can result in impressively accurate model identification. (v) Nonparametric nature of basis function models is more forgiving to structural mismatch such as input delay, model order and fast sampling rate than the conventional regression models such as ARX model (Wahlberg, 1991).

The success of orthonormal basis functions in linear system identification has motivated the extension to nonlinear system identification. In (Kalkkuhl and Katebi, 1993; Sbarbaro and Johansen, 1997), Laguerre basis functions are used for extended Wiener models. Simple delay operator sequences  $z^{-1}$  in the conventional Wiener models are replaced with Laguerre basis functions, and gate functions or local weighting functions are used to resolve the problem of excessive parameterization of conventional Wiener models. These models can also be interpreted as spatially decomposed multiple models using local weighting functions, while the linear models are replaced with Laguerre basis functions with single basis pole, to take advantage of mainly (ii),(iii) and (v) of the basis functions properties discussed above. This extended Wiener model is used for multiple model based nonlinear predictive control in (Murray-Smith and Johansen, 1997, see chap. 10). The use of orthonormal basis functions for nonlinear system identification in the existing literatures claims improved model performance for systems with unknown input delays and changing system orders. However, several limitations can be identified.

One shortcoming is that the relationship between system poles and basis functions pole (property iv) have never been exploited by fixing basis function pole to a single pole. If the system is nonlinear, it is not feasible to estimate a single basis function pole, which is a pole for a linear system. Another shortcoming is in the spatial decomposition using local weighting functions. This type of decomposition combined with simple local model structures such as linear models requires excessive number of local models to achieve reasonable accuracy and is bound to the curse of dimensionality (Narendra and Balakrishnan, 1997).

The major goal of this chapter is to extend Laguerre based identification to time varying systems, possibly rapidly changing linear systems, which may contain nonlinear systems. The basic idea is to adaptively identify Laguerre models for the corresponding operating dynamics of the system. Each Laguerre model represents different dynamics optimally, hence, this approach can represent the changing dynamics in an efficient manner. A motivating example is given in Section 2.2. In Section 2.3, Laguerre basis functions and Laguerre models are reviewed and important properties of Laguerre basis functions are illustrated. In Section 2.4, heuristic implementation of the idea is presented. More systematic algorithm is presented in Section 2.5. The proposed algorithms are compared with a fixed pole Laguerre model in Section 2.6.

## 2.2 Problem formulation

The objective is to identify a linear discrete-time, time-varying system from input-output measurements  $(u(t), y(t)), t = 0, 1, \dots$ . Specifically, we assume that the linear system is stable, well-damped and strictly proper. These assumptions are made since this study considers Laguerre basis functions, which are suitable for the approximation of stable well-damped linear systems.

The identification of this type of systems usually involves with adaptive identifica-

tion algorithms using recursive least or mean squares method, however, the variation of the system must be slow enough (Haykin, 1996). To improve the transient performance as well as convergence, multiple models method using linear local models combined with an adaptive identification can be used (Narendra et al., 1995). However, this method requires excessive number of local models and is bound to suffer from “curse of dimensionality.”

On the other hand, system identification using Laguerre model is an approximate method since Laguerre model asymptotically converge to the space that linear systems are belong to. Hence, only a finite number of Laguerre models are enough to obtain reasonable accuracy of the model and may be able to avoid the curse of dimensionality problem.

The accuracy of Laguerre models depends on three factors: (i) the order of Laguerre model, (ii) Laguerre pole and (iii) linearly parameterized weights. For adaptive filter, the factor (iii) enables the efficient estimation of weights by recursive least squares. Adaptive Laguerre models utilizing (i) have already proposed (Fejzo and Lev-Ari, 1997; Merched and Sayed, 2000). The importance of Laguerre pole is well understood in several literatures, however, the adaptation of Laguerre pole is not trivial since Laguerre pole is nonlinearly-parameterized. In (Belt and Brinker, 1996), LMS algorithm is used to realize adaptive Laguerre model with adaptive pole. However, the convergence of the model is quite slow and the model can become unstable during the adaptation.

In this work, we are motivated to realize adaptive Laguerre model to utilize the factor (ii) by multiple Laguerre models. Since Laguerre basis functions have dominance over other Laguerre basis functions, depending on the system poles, multiple Laguerre models equipped with a switching algorithm can identify rapidly changing dynamics. We conclude this subsection by giving a motivating example regarding utilizing Laguerre pole adaptation.



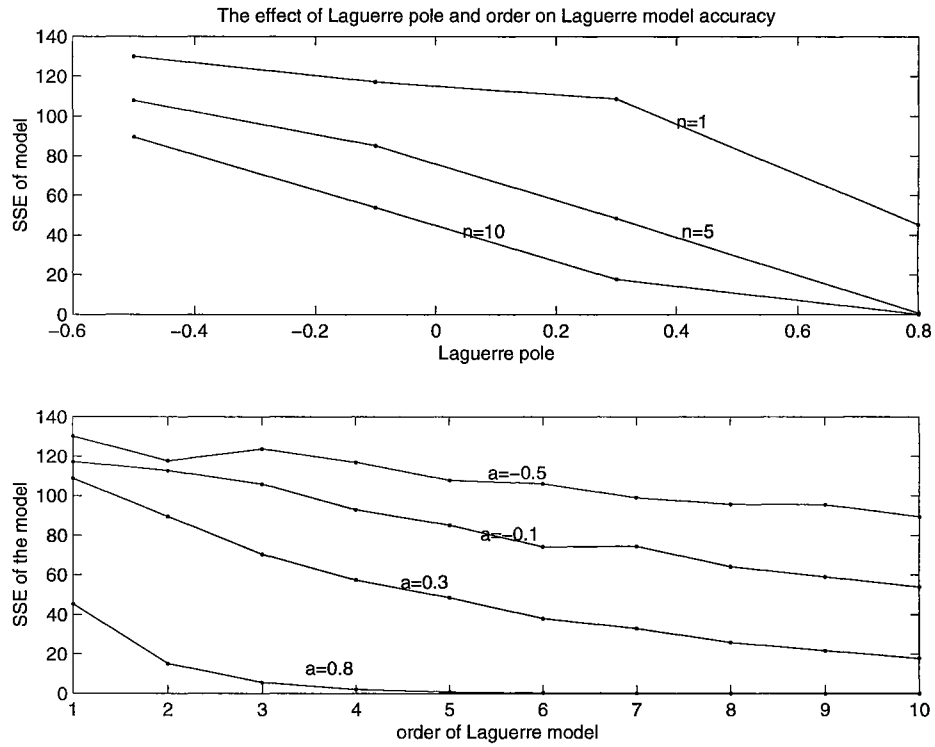


Figure 2.1: Effects of Laguerre pole and order on the accuracy of the Laguerre model: we can observe much sharper curve with respect to the Laguerre poles while the error curve become smoother and the decay of error become flat as the order of the Laguerre model increases

**Example 2.1** Consider a discrete time system

$$\frac{Y(z)}{U(z)} = \frac{1}{z - 0.9}$$

Use Laguerre models with Laguerre poles at  $\{-0.5, -0.1, 0.3, 0.8\}$  to identify the system. Vary the order of each Laguerre models from 1 to 10. Monte-Carlo simulation is performed and the averaged SSE of the model is shown in Fig. 2.1. As can be seen easily, the effects of Laguerre poles are more significant than the order of the Laguerre model.

## 2.3 Laguerre functions and Laguerre model

Laguerre basis functions and exponential functions are closely related. Exponential functions are the eigen functions of linear time invariant systems, therefore, it is

natural to use exponential functions as basis functions for the representation of linear systems. The system representation using basis functions is

$$y(t) = \left( \sum_{k=1}^{\infty} \theta_k B_k(q) \right) u(t) \quad (2.1)$$

where  $y(t)$  is output,  $u(t)$  is input,  $\{B_k(q)\}$  basis functions with forward shift operator  $q$ ,  $\{\theta_k\}$  are parameters to be estimated. For continuous time models, differential operator  $s$  replaces  $q$ .  $t$  is nonnegative integers for discrete time case and nonnegative real for continuous time case. From (2.1), impulse response is

$$h(t) = \sum_{k=1}^{\infty} \theta_k b_k(t) \quad (2.2)$$

where  $b_k(t) = B_k(q)\delta(t)$  with Kronecker delta function  $\delta(t)$ .

The analysis as well as the evaluation of optimal coefficients  $\theta_k$  will be facilitated if basis functions are orthonormal. The orthonormality of basis functions for the discrete time case can be investigated via the standard inner product on  $H_2(T)$  with  $T = \{z : |z| = 1\}$  as

$$\langle B_n, B_m \rangle = \sum_{t=0}^{\infty} b_n(t)b_m^*(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} B_n(e^{j\omega})B_m^*(e^{j\omega})d\omega \quad (2.3)$$

or the inner product as a contour integral around the unit circle  $T$  by using the change of variable  $z = e^{j\omega}$

$$\langle B_n, B_m \rangle = \frac{1}{2\pi j} \oint_T B_n(z)B_m^*(z)\frac{dz}{z} \quad (2.4)$$

where  $B^*(z)$  is the complex conjugate. For continuous time case, the standard inner product on  $H_2$  with functions analytic in the closed right half complex plane,  $Re(s) \geq 0$ , is

$$\langle B_n, B_m \rangle = \int_0^{\infty} b_n(t)b_m^*(t)dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} B_n(j\omega)B_m(j\omega)^*d\omega. \quad (2.5)$$

Then the optimal coefficients to minimize the squared error between  $h(t)$  and the approximate for discrete time system is given as

$$\theta_n = \sum_{t=0}^{\infty} h(t)b_n(t) = \frac{1}{2\pi j} \oint_T H(z)B_n^*(z)\frac{dz}{z} \quad (2.6)$$

and for continuous time system

$$\theta_n = \int_0^\infty h(t)b_n(t)dt = \frac{1}{2\pi} \int_{-\infty}^\infty H(j\omega)B_n^*(j\omega)d\omega. \quad (2.7)$$

The orthonormalization of exponential functions may be realized by the classical Gram-Schmidt process, but, it is cumbersome. Instead, a set of exponentials  $e^{-a_1t}, e^{-a_2t}, \dots, e^{-a_nt}$  with positive real exponents can be represented orthonormally by Laplace transforms as (Lai, 1985),

$$B_n(s) = \frac{\sqrt{2a_n}(s - a_1)(s - a_2)\dots(s - a_{n-1})}{(s + a_1)(s + a_2)\dots(s + a_n)}. \quad (2.8)$$

For discrete time case, similar formula is available (Ninness and Gustafsson, 1997),

$$B_n(q) = \frac{\sqrt{1 - |a_n|^2} \prod_{k=0}^{n-1} (1 - a_k q)}{q - a_n} \quad (2.9)$$

where  $|a_k| < 1$ . Laguerre basis functions are obtained from these formulae, as special cases of repeated exponents. That is, the discrete time Laguerre basis functions are defined as

$$L_n(q) = \frac{\sqrt{1 - a^2}}{q - a} \left( \frac{1 - aq}{q - a} \right)^{n-1}, n = 1, 2, \dots \quad (2.10)$$

with  $|a| < 1$ , while continuous time Laguerre basis functions are defined as

$$L_n(s) = \frac{\sqrt{2a}}{s + a} \left( \frac{s - a}{s + a} \right)^{n-1}, n = 1, 2, \dots \quad (2.11)$$

with  $a > 0$ .

Even though orthonormality of Laguerre basis functions facilitates analysis of the model, it is not so useful in system identification since it is based on the inner product in  $H_2$  sense. The major motivation of using Laguerre basis functions is their optimality in approximating analytic functions, such as transfer functions, in  $n$ -width sense. The formal definition of  $n$ -width is given as follows (Pinkus, 1985).

**Definition 2.1 (n-width measures)** Assume that the system  $G$  belongs to a given bounded set  $S$ , that is,  $G \in S$ , then  $n$ -width measure is defined as,

$$d_n(S; B) = \inf_{\Phi_n \in M_n(B)} \sup_{G \in S} \inf_{G_n \in \Phi_n} \|G - G_n\|_B \quad (2.12)$$

where  $B$  denotes some Banach space with norm  $\|\cdot\|_B$ , e.g.  $H_2$  or  $H_\infty$ ,  $M_n(B)$  denotes the collection of all  $n$ -dimensional linear subspaces of  $B$ ,  $\Phi_n = \text{span}\{B_1, B_2, \dots, B_n\}$ .

**Remark 2.1** *The inner most term,  $e_n^\Phi(G) = \inf_{G_n \in \Phi_n} \|G - G_n\|_B$  is the distance between the system  $G$  and  $\Phi_n$ , a subspace spanned by  $n$  bases. Hence,  $n$ -width measures the smallest approximation error, for the worst possible system in a given system set, using the best possible  $n$ -dimensional subspace.*

The optimality of Laguerre basis functions is proved by utilizing the optimality of FIR models for exponentially stable discrete-time system. First, introduce the optimality of FIR model.

**Theorem 2.1** ((Mäkilä and Partington, 1993)) *Let  $R \geq 1$ ,  $K > 0$ ,  $m = 0, 1, \dots$  then*

$$d_n(A_R(m, K); H_p(|z| < R)) = \begin{cases} \infty & \text{if } n < m, \\ KR^{m-n}(n-m)!/n! & \text{if } n \geq m \end{cases} \quad (2.13)$$

and  $\text{span}\{1, z, z^2, \dots, z^{n-1}\}$  is an optimal subspace for  $d_n(A_R(m, K); H_p(|z| < R))$ , where  $A_R(m, K) = \{G : G \in H_p(|z| < R), \|G^{(m)}\|_{H_p(|z| < R)} < K\}$ . Here,  $H_p(|z| < R)$  denote the Hardy space of bounded analytic functions in an open disk of radius  $R > 0$ .

Theorem 2.1 is proved for a system analytic inside  $|z| < R$ . However, a stable linear system is analytic outside of the disk  $|z| < R$  for  $R \leq 1$  or in  $|z| > R$ . Hence, the theorem can be modified by replacing  $z$  with  $w^{-1}$  and we can easily derive the following corollary. We consider  $m = 0$  as a special case for later use.

**Corollary 2.1 (Optimality of causal FIR model)** *Let  $R \leq 1$ ,  $K > 0$  then*

$$d_n(A'_R(0, K); H_p(|w| > R)) = KR^n \quad (2.14)$$

and  $\text{span}\{1, w^{-1}, w^{-2}, \dots, w^{-(n-1)}\}$  is an optimal subspace for  $d_n(A'_R(0, K); H_p(|w| > R))$ , where  $A'_R(0, K) = \{G : G \in H_p(|w| > R), \|G\|_{H_p(|w| > R)} < K\}$ . Here,  $H_p(|w| > R)$

$R$ ) denote the Hardy space of bounded analytic functions, that is, analytic function and Lebesgue integrable along the boundary of an open disk of radius  $R > 0$ .

Based on Theorem 2.1 and the bilinear transformation  $w = (z - a)/(1 - az)$  mapping the disk  $D(c, r)$ , that is centered at  $c$  with radius of  $r$  in the  $z$ -plane conformally onto the disk  $D(0, R)$  in the  $w$ -plane, the optimality of rational basis function is proved in the following Theorem.

**Theorem 2.2 ((Mäkilä and Partington, 1993))** *Let  $K > 0$ ,  $c \neq 0$  and  $r > |c| + 1$  be given. Then*

$$d_n(A_r^c(K); H_\infty(|z - c| < r)) = K \left[ \frac{r^2 - (a - c)^2}{(1 - ac)^2 - r^2 a^2} \right]^{-n/2}, n \geq 0 \quad (2.15)$$

where

$$a = \frac{1}{2c}(1 + c^2 - r^2 + [(1 + c^2 - r^2)^2 - 4c^2]^{1/2}) \quad (2.16)$$

so that  $0 < |a| < 1$ . Furthermore,

$$\text{span} \left\{ 1, \left( \frac{z - a}{1 - az} \right), \dots, \left( \frac{z - a}{1 - az} \right)^{n-1} \right\} \quad (2.17)$$

is an optimal subspace for  $d_n(A_r^c(K); A(D))$ . Here

$$A_r^c(K) = \{G : G \in H_\infty(D(c, r)), \|G\|_{H_\infty(D(c, r))} \leq K\}.$$

**Remark 2.2** *Theorem 2.2 in (Mäkilä and Partington, 1993) stated in terms of  $H_\infty$  norm. But, the results hold for general  $p$ -norm including  $H_2$ -norm (Pinkus, 1985).*

Similarly, using the Corollary 2.1 and the bilinear transformation  $w = (z - a)/(1 - az)$ , we can derive more useful result for stable linear systems. The following corollary lists the result.

**Corollary 2.2 (Optimality of stable Laguerre basis functions)** *Let  $K > 0$ ,  $c \neq 0$ ,  $r + |c| < 1$ , then*

$$d_n(A_r^c(K); H_2(|z - c| > r)) = KR^n, n \geq 0 \quad (2.18)$$

where

$$a = \frac{1}{2c}(1 + c^2 - r^2 - [(1 + c^2 - r^2)^2 - 4c^2]^{1/2}) \quad (2.19)$$

such that  $|a| < 1$  and

$$\text{span} \left\{ \frac{\sqrt{1-a^2}}{z-a}, \frac{\sqrt{1-a^2}}{z-a} \left( \frac{1-az}{z-a} \right), \dots, \frac{\sqrt{1-a^2}}{z-a} \left( \frac{1-az}{z-a} \right)^{n-1} \right\} \quad (2.20)$$

is the optimal subspace. Here,

$$R = \left[ \frac{r^2 - (c-a)^2}{(1-ac)^2 - r^2 a^2} \right]^{1/2} \quad (2.21)$$

and

$$A_r^c(k) = \{G \in H_2(|z-c| > r) : \|G\|_{H_2(|z-c| > r)} \leq K\} \quad (2.22)$$

**Proof:** The bilinear transformation

$$w = (z-a)/(1-az) \quad (2.23)$$

is a special form of the most general bilinear transformation  $w = k(z-a)/(\bar{a}z-1)$  that maps the disk  $|z| < 1$  onto the disk  $|w| < 1$ , with the arbitrary point  $a$  in the disk  $|z| < 1$  to the center of the disk  $|w| = 1$ , where  $|k| = 1$  and  $|a| < 1$  (Jeffrey, 1992, Theorem 2.10).

We use this bilinear transformation to determine  $a$  such that  $|a| < 1$  and map  $|z-a| < r$ ,  $r+|c| < 1$  onto  $|w| < R$ ,  $R < 1$ . If  $a$  is real, (2.23) maps real points in  $z$  onto real points in  $w$ -plane. The choice of  $a$  will move the center of the circle in the  $w$ -plane along the real axis in that plane. So,  $a$  must be chosen to map  $c+r$  and  $c-r$  symmetric to the origin in  $w$ -plane, that is,

$$\frac{c+r-a}{1-a(c+r)} = -\frac{c-r-a}{1-a(c-r)}. \quad (2.24)$$

Solve for  $a$  and we obtain

$$a = \frac{1}{2c}[(1+c^2-r^2) \pm \sqrt{(1+c^2-r^2)^2 - 4c^2}]. \quad (2.25)$$

Now choose  $a$  such that  $|a| < 1$ . Since we assume that  $r + |c| < 1$ ,  $a$  is real. Consider the positive case first. If

$$\left| \frac{1}{2c} [(1 + c^2 - r^2) + \sqrt{(1 + c^2 - r^2)^2 - 4c^2}] \right| < 1 \quad (2.26)$$

then

$$(1 + c^2 - r^2) + \sqrt{(1 + c^2 - r^2)^2 - 4c^2} < |2c|. \quad (2.27)$$

since the l.h.s is positive because  $r + |c| < 1$  and equivalently  $-1 < c - r < c + r < 1$ .

For  $c > 0$ ,

$$\sqrt{(1 + c^2 - r^2)^2 - 4c^2} < 2c - (1 + c^2 - r^2) \quad (2.28)$$

and

$$(c - 1)^2 - r^2 = (c - 1 + r)(c - 1 - r) < -\sqrt{(1 + c^2 - r^2)^2 - 4c^2} \quad (2.29)$$

Since  $c - r < c + r < 1$ , the left hand side is positive and it contradicts since right hand side is negative. Hence, it can be easily seen that

$$a = \frac{1}{2c} [(1 + c^2 - r^2) - \sqrt{(1 + c^2 - r^2)^2 - 4c^2}]. \quad (2.30)$$

The radius of disk  $|w| > R$  is

$$R^2 = w\bar{w} = - \left[ \frac{c + r - a}{1 - a(c + r)} \right] \left[ \frac{c - r - a}{1 - a(c - r)} \right]. \quad (2.31)$$

Therefore,

$$R = \left[ \frac{r^2 - (c - a)^2}{(1 - ac)^2 - r^2 a^2} \right]^{1/2}. \quad (2.32)$$

Now, use the Corollary 2.1 and we can conclude that

$$d_n(A_r^c(K); H_2(|z - c| > r)) = KR^n, \quad n \geq 0 \quad (2.33)$$

for

$$A_r^c(k) = \{G \in H_2(|z - c| > r) : \|G\|_{H_2(|z - c| > r)} \leq K\}. \quad (2.34)$$

and the optimal subspace is

$$\text{span} \left\{ 1, \left( \frac{1-az}{z-a} \right), \dots, \left( \frac{1-az}{z-a} \right)^{n-1} \right\} \quad (2.35)$$

However, the bilinear transformation changes the  $H_2$ -norm since the corresponding measure is changed as

$$\frac{dw}{w} = \frac{1-a^2}{(1-az)(z-a)} \frac{dz}{z} \quad (2.36)$$

$$= \frac{\sqrt{1-a^2}}{z-a} \frac{\sqrt{1-a^2}}{1/z-a} \frac{dz}{z}. \quad (2.37)$$

Hence, the inner product is defined as

$$\langle G_n, G_m \rangle = \frac{1}{2\pi j} \oint_T G_n(w) G_m(1/w) \frac{dw}{w} \quad (2.38)$$

$$= \frac{1}{2\pi j} \oint_T H_n(z) H_m(1/z) \frac{\sqrt{1-a^2}}{z-a} \frac{\sqrt{1-a^2}}{1/z-a} \frac{dz}{z}. \quad (2.39)$$

Therefore, basis functions must be multiplied by  $\sqrt{1-a^2}/(z-a)$ . This implies that

$$\text{span} \left\{ \frac{\sqrt{1-a^2}}{z-a}, \frac{\sqrt{1-a^2}}{z-a} \left( \frac{1-az}{z-a} \right), \dots, \frac{\sqrt{1-a^2}}{z-a} \left( \frac{1-az}{z-a} \right)^{n-1} \right\} \quad (2.40)$$

is the optimal subspace.  $\square$

**Remark 2.3** *This corollary has been commented in several literatures and the proof is illustrated in (Wahlberg, 1994, 1999). However, no formal proof is given and often the  $n$ -width measure is not considered. We presented the complete results scattered in the literatures and make a correction on the optimal pole equation in this corollary.*

## 2.4 Heuristic adaptive multiple Laguerre models

From the previous section, it is obvious that the optimal pole of Laguerre function and the weighting parameters change as the system changes. While the estimation of weighting parameters is trivial since it is linear-in-the-parameters structure, determination of Laguerre pole requires to solve a nonlinear optimization problem. In



literatures, the difficulty of estimating optimal poles of orthonormal basis functions is clearly illustrated. In (Fu and Dumont, 1993), analytical solution for optimal Laguerre pole for discrete time Laguerre basis functions is given, when the impulse response of the system is available. To facilitate the determination of Laguerre pole, the condition of optimal Laguerre pole for truncated Laguerre models is given in (Wang and Cluett, 1994) for continuous time case and in (Masnadi-Shirazi and Ahmed, 1991) for discrete time case when the input to the system is impulse. Optimality condition for truncated Laguerre model for an arbitrary input case is given in (Oliveira e Silva, 1994). It is interesting to see that the optimality condition for truncated Laguerre model is

$$\theta_n(a)\theta_{n+1}(a) = 0 \quad (2.41)$$

for truncated  $n$ th order Laguerre models for all the cases. In (Oliveira e Silva, 1995), this optimality condition is used for the estimation of optimal Laguerre pole. However, this work is only limited for local search since interpolation/extrapolation functions based on series approximation of  $\theta(a)$  is used to solve for the optimality condition. Hence, there is no clear advantage of utilizing the analytically obtained optimality conditions for system identification purpose.

Rather, the optimal Laguerre pole is found by solving nonlinear programming. In (Sabatini, 2000), genetic algorithm in the combination with gradient based algorithm is used to search for the global optimal Laguerre pole for impulse response case. We present Newton-Raphson's iteration method for discrete time Laguerre pole estimation for an arbitrary input in Section 2.4.1. Similar derivation of the analytical Gradient and Hessian is presented in (Malti et al., 1998). Their derivation is based on Laguerre states and matrix structure while the presented derivation is by utilizing the filter structure for the easy of MATLAB implementation.

### 2.4.1 Optimal Laguerre pole estimation

The  $m$ -th order truncated Laguerre model is represented as

$$\hat{y}(t) = G(q)u(t) = \sum_{i=1}^m \theta_i L_i(q)u(t) \quad (2.42)$$

where  $L_i(q)$  is the  $i$ th Laguerre function, that is,  $L_i(q) = \frac{\sqrt{1-a^2}}{q-a} \left(\frac{1-aq}{q-a}\right)^{i-1}$ ,  $i = 1, \dots, m$  and  $|a| < 1$ .  $a$  is the pole of Laguerre basis. Consider summed squared cost function

$$J = \frac{1}{2} E^T E \quad (2.43)$$

where  $E = Y - \hat{Y} = Y - X\theta$ ,  $Y = [y(1), y(2), \dots, y(N)]^T$ ,

$$X = [u(1), u(2), \dots, u(N)]^T [L_1(q), L_2(q), \dots, L_m(q)] = U [L_1(q), L_2(q), \dots, L_m(q)]$$

and  $\theta = [\theta_1, \theta_2, \dots, \theta_m]^T$ .

The modified Newton-Raphson's method is:

$$a_{n+1} = a_n - \mu_n \left( \frac{\partial^2 J(a_n)}{\partial a^2} \right)^{-1} \frac{\partial J(a_n)}{\partial a} \quad (2.44)$$

where  $\partial J(a_n)/\partial a$  and  $\partial^2 J(a_n)/\partial a^2$  are the first and second partial derivative of  $J$  with respect to  $a$ , evaluated at  $a_n$ .  $\mu_n$  is the step size. The derivative of  $J$  with respect to  $a$  is

$$\frac{\partial J}{\partial a} = E^T \frac{\partial E}{\partial a} = E^T \left( -\frac{\partial X}{\partial a} \theta - X \frac{\partial \theta}{\partial a} \right). \quad (2.45)$$

The parameter vector  $\theta$  can be written in a closed form solution, the so called normal equation:

$$\theta = (X^T X)^{-1} X^T Y. \quad (2.46)$$

The partial derivative of the normal equation can be written as:

$$\frac{\partial \theta}{\partial a} = (X^T X)^{-1} \left[ \frac{\partial X^T}{\partial a} Y - \left( \frac{\partial X^T}{\partial a} X + X^T \frac{\partial X}{\partial a} \right) \theta \right]. \quad (2.47)$$

The second term of (2.45),  $E^T X \frac{\partial \theta}{\partial a}$ , becomes zero since  $(X^T X)^{-1} X^T E$  is zero because of the well known orthogonality property of Least square estimation. Hence, (2.45)

simply reduces to

$$\frac{\partial J}{\partial a} = -E^T \frac{\partial X}{\partial a} \theta. \quad (2.48)$$

The second derivative of  $J$  with respect to  $a$  is:

$$\frac{\partial^2 J}{\partial a^2} = \left( \theta^T \frac{\partial X^T}{\partial a} + \frac{\partial \theta^T}{\partial a} X^T \right) \frac{\partial X}{\partial a} \theta - E^T \left( \frac{\partial^2 X}{\partial a^2} \theta + \frac{\partial X}{\partial a} \frac{\partial \theta}{\partial a} \right). \quad (2.49)$$

So, the remaining equations to be derived are the first and second partial derivatives of  $X$ . The derivation is done by the following Laguerre filter property (Wang and Cluett, 1994):

$$\frac{\partial L_i}{\partial a} = \frac{iL_{i+1}(a) - (i-1)L_{i-1}(a)}{1-a^2}, i = 1, 2, \dots, m. \quad (2.50)$$

Therefore, the first partial derivative of  $X$  is:

$$\frac{\partial X}{\partial a}(:, i) = U \left( \frac{iL_{i+1}(a) - (i-1)L_{i-1}(a)}{1-a^2} \right), i = 1, 2, \dots, m \quad (2.51)$$

where  $(:, i)$  means the  $i$ -th column of a matrix.

The second partial derivative of  $X$  is the following:

$$\begin{aligned} \frac{\partial^2 X}{\partial a^2}(:, i) = U & \left( \frac{i(i+1)L_{i+2}(a) + 2aiL_{i+1}(a) - (2i^2 - 2i + 1)L_i(a)}{(1-a^2)^2} \right. \\ & \left. + \frac{-2a(i-1)L_{i-1}(a) + (i-1)(i-2)L_{i-2}(a)}{(1-a^2)^2} \right), i = 1, 2, \dots, m. \end{aligned} \quad (2.52)$$

The algorithm is a combination of linear (2.46) and nonlinear least squares method (2.44), that is the so called separable least squares (Ljung, 1999). This method gives numerically well-conditioned calculations.

Since the objective function (2.43) is highly nonlinear multimodal function of the Laguerre pole  $a$  as shown in (Oliveira e Silva, 1994), the initial  $a$  must be close to the global minimum. Several trials are necessary to obtain a satisfying estimate of  $a$ .

## 2.4.2 Heuristic implementation by recursive LS with multiple Laguerre models

The numerical search algorithm derived in the previous section is a batch method. It is certainly not impossible to use the algorithm for adaptive pole estimation, however,

the multimodality of the objective function with respect to Laguerre pole can cause severe problems when the solution converges to local minima. Indeed, the estimation of proper Laguerre poles for a given i/o measurements must be repeated several times with different initial conditions, combined with visual inspection to verify the quality of the estimated Laguerre poles.

Instead, we use the pole estimation off-line only using the training data by segmenting the data set and apply the numerical search at each data segment. The estimated pole at each data segment is compared with the estimated poles with the previous data segments and is accepted as new Laguerre pole if the difference is reasonably large. Multiple Laguerre models can be generated in this manner.

For the structural adaptation of linear models, additional terms to improve the model accuracy are ordered. Therefore, there is no problem to choose the best additional term from overwhelming combinations of possible terms. The online implementation of multiple Laguerre models is not trivial since Laguerre models are not ordered even if individual Laguerre model has ordered terms.

In this section, our approach is simply to utilize recursive least squares to update parameters of all the identified Laguerre models off-line without on-line model selection process. The well-known recursive least square with forgetting factor is the following:

$$\begin{aligned}\theta_{k+1} &= \theta_k + \alpha_k P_k X(k+1)(Y(k+1) - X(k+1)^T \theta_k) & (2.53) \\ \alpha_k &= \frac{1}{\lambda + X(k+1)^T P_k X(k+1)} \\ P_{k+1} &= \frac{P_k - P_k X(k+1) \alpha_k^{-1} X(k+1)^T P_k}{\lambda}\end{aligned}\tag{2.54}$$

where  $\lambda$  is a forgetting factor,  $X(k+1)$  is a regression vector at  $k+1$  sequence,  $Y(k+1)$  is a output at  $k+1$  sequence and  $\theta_k$  is the estimated parameter at  $k$  sequence.  $P_k$  is a initially large number.

## 2.5 Adaptive multiple Laguerre models using recursive orthogonal LS with Laguerre poles in irregular grid

Since estimation of optimal Laguerre poles by nonlinear programming can be problematic because of multimodality of the cost function with respect to Laguerre pole, and Laguerre poles are only distributed in a real axis in the range of  $|a| < 1$ , determination of Laguerre poles in grid is feasible. One intuitive idea can be regular grid, however, this may requires more local models than irregular grid utilizing data or model properties. In addition, the performance of Laguerre models depends on two factor: the order of Laguerre model and Laguerre pole. Therefore, regular grid may require different model orders for different Laguerre models. In this section, we propose multiple Laguerre models with irregular grid utilizing the approximation property of Laguerre functions. Recursive orthonormal least squares method is used to recursively estimate parameters of the model as well as to identify dominant Laguerre models from incoming i/o measurements.

### 2.5.1 Multiple Laguerre models using irregular grid

From Corollary 2.2, we see that the Laguerre basis model of a proper Laguerre pole can be superior to other Laguerre models, given a system with poles inside a disk. This is the motivation of using multiple Laguerre models with different poles. In this subsection, we utilize Corollary 2.2 to obtain Laguerre poles distributed in irregular grids.

For a finite dimensional system  $G_0(z)$  with simple poles  $p_j$ , the asymptotic decay rate of the impulse response is defined as  $\max_j |p_j|$ . From Corollary 2.2,  $R$  is the boundary of the disk centered at the origin. Since Laguerre basis functions are transformed to the disk with center at the origin,  $R$  can represent the slowest decay rate of the Laguerre model in the transformed domain, therefore, is a good indicator of

the performance of Laguerre models. Hence, we utilize this factor to obtain irregular grids.

From Corollary 2.2,

$$R = \frac{r^2 - (a - c)^2}{(1 - ac)^2 - r^2 a^2} \quad (2.55)$$

with the Laguerre optimal pole for a disk  $|z - c| < r$

$$a = \frac{1}{2c}(1 + c^2 - r^2 - [(1 + c^2 - r^2)^2 - 4c^2]^{1/2}). \quad (2.56)$$

We want to cover  $-1 < \text{real}(z) < 1$  of the real axis of  $z$ -domain with finite number of open disk  $|z - c_i| < r_i$ ,  $i \in \mathbb{N}$  with corresponding Laguerre poles for each disk. The algorithm is simply to generate disks sequentially starting from the origin while solving nonlinear equations (2.55) and (2.56) for  $c_i$ ,  $r_i$  and  $a_i$  where  $i$  denotes the  $i$ th Laguerre model and  $i \in \{1, \dots, n_p, n_p + 1, \dots, n_p + n_n\}$  where  $n_p$  and  $n_n$  implies the number of Laguerre poles at the positive and negative real axis, respectively.

Experimental works show that solving nonlinear equations (2.55) and (2.56) is problematic near 1 and  $-1$ , because of numerical problems. One remedy can be to cover the regions near these points with disks with decaying radius as they approach the unity points. Hence, the reduction rate of the disks  $\beta$  has the effect of  $r_{i+1} = \beta r_i$ .

Another problem is caused by the condition that the Laguerre basis functions have to satisfy, that is,  $|a_i| < 1$ . The disk is to cover the last circle 1 or  $-1$  at the boundary results in  $a = 1$ . This is remedied by placing some small margin such that the boundary of the last circle does not contain the unity point. This factor is controlled by the variable  $\gamma$  in the algorithm. The irregular grid generation is illustrated in Fig. 2.2. The algorithm is listed as follows.

**Design parameter**  $R$  in the range of  $0 < R < 1$ ,  $\beta$  and  $\gamma$

**step 1** For  $i = 1$ , set  $c_1 = r_1$  and  $a_1$  with (2.56). Solve (2.55) for  $c_1$ . Update  $r_1 = c_1$  and  $a_1$  by (2.56).

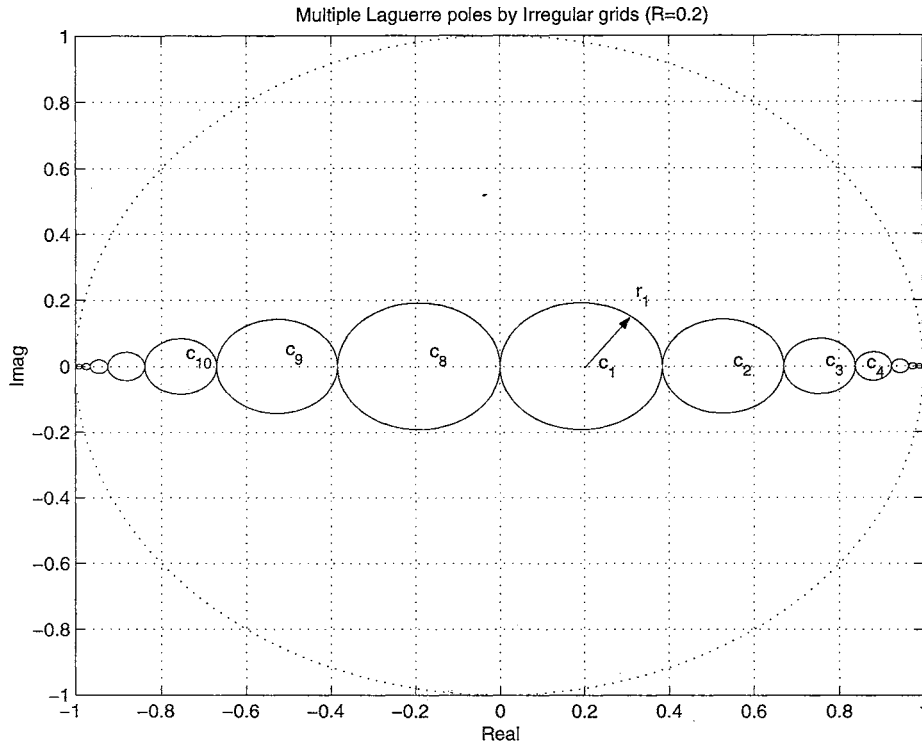


Figure 2.2: Irregular grid for multiple Laguerre models.

**step 2** For  $i \geq 2$ , let  $c_i = r_i + 2 \sum_{j=1}^{i-1} r_j$  and  $a_i$  with (2.56). Solve the nonlinear equation (2.55).

**step 3** Check if  $|c_i| + r_i < 1$ . If it is, update  $r_i = c_i - 2 \sum_{j=1}^{i-1} r_j$  and  $a_i$  by (2.56) and continue with step 2. If not or solving the nonlinear equations fails, go to step 4.

**step 4** Generate disk with radius  $r_i = \beta r_{i-1}$ , center at  $c_i = r_i + 2 \sum_{j=1}^{i-1} r_j$  until  $2 \sum_{j=1}^i r_i < 1 - \gamma$ . Generate the last disk by  $r_n = (1 - \gamma - \sum_{j=1}^{n-1} r_j)/2$  and  $c_n = r_n + 2 \sum_{j=1}^{n-1} r_j$ .

**step 5** Repeat step 1 through 4 on the negative axis with  $c_1 = -r_{n_p+1}$  in step 1,  $c_i = -(2 \sum_{j=n_p+1}^{i-1} r_j + r_i)$  in step 2,  $r_i = -c_i - 2 \sum_{j=n_p+1}^{i-1} r_j$  in step 3, and  $c_i = -(r_i + 2 \sum_{j=n_p+1}^{i-1} r_j)$  and  $c_{n_p+n_n} = -(r_{n_p+n_n} + 2 \sum_{j=n_p+1}^{n_p+n_n-1} r_j)$  in step 4.

**Remark 2.4** *The covering of negative real axis is indeed symmetric to the positive*

real axis. The covering of negative real axis with different  $R$  may sometimes be necessary to reduce oscillation of model output.

## 2.5.2 Review of orthogonal least squares methods

Least squares estimation is commonly used to estimate model parameters for linear models. This method has advantages of fast estimation and the existence of closed form solution to facilitate analysis of the estimate. The classical least squares method is a batch form while the recursive form can be obtained from matrix inversion lemma in a straight forward manner (Mendel, 1995). One limitation of recursive least squares in this form is that the model structure, that is, the number of parameters (or the order of the model) is fixed and inclusion of more model parameters requires starting over the estimation.

Orthogonalization of the estimate can circumvent this problem. For models that have recursive structures, efficient order update algorithm has already been developed based on orthogonalization of least squares estimate by the so called lattice structure (Haykin, 1996; Merched and Sayed, 2000). Orthogonal least squares method is still beneficial in non-lattice structures since efficient QR decomposition can provide better numerical stability than the conventional recursive LS estimate. We give a brief review of orthogonal least squares in this subsection. The presentation follows (Stark, 1997).

Least square estimate is used to solve linear equations such as

$$A\theta = Y \tag{2.57}$$

to minimize

$$\|A\theta - Y\|^2 = (A\theta - Y)^T(A\theta - Y) \tag{2.58}$$

where  $A$ ,  $Y$  and  $\theta$  are matrices with  $N \times m$ ,  $N \times 1$  and  $m \times 1$  sizes. Orthogonalization of  $A$  matrix can be done by QR decomposition as  $A = QR$  where  $Q$  is a  $N \times m$



matrix with orthogonal columns and  $R$  is a  $m \times m$  upper triangular matrix. Define an augmented matrix  $\tilde{A} = [A, Y]$  then the decomposition  $\tilde{A} = \tilde{Q}\tilde{R}$ . If we write  $\tilde{Q} = [Q, q]$  and

$$\tilde{R} = \begin{bmatrix} R & r \\ 0 & \tilde{r} \end{bmatrix} \quad (2.59)$$

then  $\tilde{A}$  matrix is

$$\tilde{A} = \tilde{Q}\tilde{R} = [Q \quad q] \begin{bmatrix} R & r \\ 0 & \tilde{r} \end{bmatrix}. \quad (2.60)$$

Then the square of norm of residue can be rewritten as

$$\|A\theta - Y\|^2 = (QR\theta - Qr - q\tilde{r})^T(QR\theta - Qr - q\tilde{r}) \quad (2.61)$$

$$= \|Q(R\theta - r)\|^2 + \tilde{r}^2\|q\|^2 \quad (2.62)$$

by utilizing the orthogonality of columns of  $\tilde{Q}$  matrix. Then, the least squares solution is  $R\theta - r = 0$  and this can be easily computed by methods such as back-substitution.

To show the recursive QR least squares, we present  $a(t+1)$ , a new row of the matrix  $A$  and a new value  $y(t+1)$ . Define  $A(t) = [a(1)^T, a(2)^T, \dots, a(t)^T]^T$  and  $y(t) = [y(1), \dots, y(t)]^T$ . Then  $\tilde{A}$  matrix with an additional row is

$$\tilde{A}(t+1) = \begin{bmatrix} \tilde{A}(t) \\ \tilde{a}(t+1) \end{bmatrix} = \begin{bmatrix} \tilde{Q}(t)\tilde{R}(t) \\ \tilde{a}(t+1) \end{bmatrix} = \begin{bmatrix} \tilde{Q}(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{R}(t) \\ \tilde{a}(t+1) \end{bmatrix} = \bar{Q}(t+1)\bar{R}(t+1). \quad (2.63)$$

The orthogonal decomposition of  $\bar{R}(t+1)$  is

$$\bar{R}(t+1) = \bar{Q}'(t+1)\tilde{R}'(t+1) \quad (2.64)$$

and then  $\tilde{A}(t+1) = \tilde{Q}(t+1)\tilde{R}(t+1)$  where  $\tilde{Q}(t+1) = \bar{Q}(t+1)\bar{Q}'(t+1)$  and  $\tilde{R}(t+1) = \bar{R}'(t+1)$ . Notice that  $\tilde{Q}(t+1)$  is again orthogonal and  $\tilde{R}(t+1)$  is upper triangular, which results into a QR decomposition again.

The nice orthogonal property of QR least squares facilitates the error analysis. Let  $m_s$  be the columns of  $A$  that are selected and ignore the remaining  $m - m_s$  columns. Then  $A = [A_s, A_d]$  where  $A_d$  is the deselected columns. Similarly, partitioning  $Q =$

$[Q_s, Q_d]$  and the orthogonal decomposition of  $\tilde{A}$  becomes

$$[A_s, A_d, Y] = [Q_s, Q_d, q] \begin{bmatrix} R_s & R_{sd} & r_s \\ 0 & R_d & r_d \\ 0 & 0 & \tilde{r} \end{bmatrix} \quad (2.65)$$

and similarly with (2.62) we can write

$$\|A\theta - Y\|^2 = \|Q_s(R_s\theta_s + R_{sd}\theta_{sd} - r_s)\|^2 + \|Q_d(R_d\theta_d - r_d)\|^2 + \tilde{r}^2\|q\|^2. \quad (2.66)$$

If we ignore the last  $m - m_s$  columns,  $\theta_d = 0$  and solve the least square equation, which results in

$$\|A\theta - Y\|^2 = \|Q_d r_d\|^2 + \tilde{r}^2\|q\|^2 \quad (2.67)$$

and for the orthonormal  $Q$ , we have

$$\|A\theta - Y\|^2 = \sum_{i=m_s+1}^m r_i^2 + \tilde{r}^2. \quad (2.68)$$

Notice that for error computation as well as LS solution, we only require the fixed sized upper triangular matrix,  $R$ . This is an important property of orthogonal LS estimate to be used for recursive estimation.

By permutating the columns of  $A$ , we may partition the matrix into  $[A_s, A_d]$  easily. The permutation does not change the LS solution  $Y = A_1\theta_1 + A_2\theta_2$ , since the permutation of the columns of  $A$  matrix, e.g.

$$\tilde{A} = [A_1, A_2, Y] = [Q_1, Q_2, q] \begin{bmatrix} R_1 & R_2 & r_1 \\ 0 & R_3 & r_2 \\ 0 & 0 & \tilde{r} \end{bmatrix} \quad (2.69)$$

results in

$$\tilde{A}\Pi = [A_2, A_1, Y] = [Q_1, Q_2, q] \begin{bmatrix} R_2 & R_1 & r_1 \\ R_3 & 0 & r_2 \\ 0 & 0 & \tilde{r} \end{bmatrix} = Q'R' \quad (2.70)$$

where  $\Pi$  is a permutation matrix that switch columns.  $R'$  can be QR decomposed again by  $R' = Q_2R$ , hence,  $Q'R' = Q'Q_2R$  is again orthogonalized form. The LS estimation of the permuted matrix is to solve

$$\begin{bmatrix} R'_1 & R'_2 \\ 0 & R'_3 \end{bmatrix} \begin{bmatrix} \theta_2 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \quad (2.71)$$

where  $R'_1$ ,  $R'_2$  and  $R'_3$  are block matrices of  $R'$  matrix as (2.69). Hence, the permutation  $A$  requires only the permutation of  $R$ , reorthogonalization and the LS solution. This results in the permutation of corresponding rows of  $\theta$ .

### 2.5.3 Adaptive Laguerre model selection using orthogonal least squares methods

The review of orthogonal LS estimation clearly shows its advantage in the recursive application of the estimation method. The time update of the estimate only requires updating the smaller fixed sized upper triangular matrix  $R$ . The order update is equivalent to adding additional columns in the  $R$  matrix for increasing the order, and deleting or ignoring existing columns for decreasing. From (2.68), we can easily compute the residue of the estimated model with selected parameters, which can be used for selecting the proper subspace of the model efficiently.

In order to utilize the orthogonal LS recursively for adaptive multiple Laguerre models, we need to consider several factors. Since individual Laguerre models are not ordered, the best or proper Laguerre models must be selected based on the residue of individual models. This gives rise to huge combinational problem. This problem has been the major issue in nonlinear system identification using radial basis functions (Stark, 1997; Chen et al., 1991). These works are concentrated on relieving the computational load by computing the change of error by keeping or removing individual basis without estimating the model parameters.

One difference between multiple Laguerre models in this study and radial basis functions is that individual Laguerre model is consist of Laguerre basis functions with certain order while basis of radial basis functions is completely independent. Therefore, the  $A$  matrix for multiple Laguerre models actually consists of block columns that correspond to each Laguerre model characterized by different Laguerre poles.

Hence,  $A = [A_1, A_2, \dots, A_m]$  and  $A_i$  consists of

$$A_i(t) = \begin{bmatrix} x_1^{(i)}(0) & x_2^{(i)}(0) & \dots & x_n^{(i)}(0) \\ \vdots & & & \vdots \\ x_1^{(i)}(t) & x_2^{(i)}(t) & \dots & x_n^{(i)}(t) \end{bmatrix} \quad (2.72)$$

where  $x_k^{(i)}(t)$  is the  $k$ th state of the  $i$ th Laguerre model at time  $t$  and  $i = 1, \dots, m$  for  $m$ -multiple Laguerre models. The Laguerre states can be recursively calculated as

$$x_k^{(i)}(t) = \begin{cases} a^{(i)}x_k^{(i)}(t-1) + \sqrt{1-a^{(i)2}}u(t-1) & \text{if } k = 1 \\ a^{(i)}x_k^{(i)}(t-1) - a^{(i)}x_{k-1}^{(i)}(t) + x_{k-1}^{(i)}(t-1) & \text{for } 2 \leq k \leq n \end{cases} \quad (2.73)$$

where  $n$  is the order of a Laguerre model.

Another difference is in that in radial basis functions, the goal is to add or remove individual basis sequentially. In multiple Laguerre models, the goal is to identify the best combination of Laguerre models at each data sequence. For  $m$  Laguerre models, the possible combination is

$$\sum_{k=1}^m \frac{m!}{(m-k)!k!} \quad (2.74)$$

One problem of using residue to select the best model is that the residue is a decreasing function of the number of models. This is because auxiliary models are fitted to noise or insignificant information. As a result, the best combination of models that generate the smallest error is the one with largest size. Hence, the proper model selection method is to find models at the knee of the decreasing error curve with respect to number of models. However, generating the error curve and selecting the proper model subset are not feasible for online applications. We handle this problem by setting the upper limit in the number of Laguerre models that can exist concurrently. By setting this limit, the combinational cases reduce to

$$\frac{m!}{(m-n_m)!n_m!} \quad (2.75)$$

where  $n_m$  is the upper limit of the number of concurrent Laguerre models.

The second issue to consider is the forgetting factor. Since recursive LS estimate is equivalent to the batch LS estimate without forgetting factor, the changing dynamics

cannot be identified without forgetting factor. Indeed, forgetting factor is the key component to decide the performance of recursive LS estimate. It is intuitive that the forgetting factor has to be large for rapidly changing dynamics, while it has to be small to obtain unbiased estimate for stationary signals. In (Andersson, 1985), adaptive forgetting algorithm is derived by multiple Kalman filters based on the assumption of known probability of parameter jumps. Since we only consider deterministic case in this study, we use a simple adaptation rule of the forgetting factor such as

$$\lambda = 1 - y_\lambda(t) \quad (2.76)$$

where  $\lambda$  is the forgetting factor, and  $y$  is an adaptation variable defined as

$$y_\lambda(t) = a_\lambda y_\lambda(t-1) + b_\lambda e(t-1)^2. \quad (2.77)$$

$a_\lambda$  and  $b_\lambda$  can be used as tuning variables.  $e(t-1)$  is the residue at time  $t-1$ . Notice that  $y(t) \geq 0$  at all  $t$ . Since too small  $\lambda$  is undesirable, it is safe to set some upper limit on  $y(t)$ . Hence, the resulting adaptation variable becomes

$$y_\lambda(t) = \begin{cases} a_\lambda y_\lambda(t-1) + b_\lambda e(t-1)^2 & \text{if } y_\lambda(t) < y_{\lambda u} \\ y_{\lambda u} & \text{otherwise} \end{cases}. \quad (2.78)$$

Because of noise in the measurement or numerical errors, improperly fast switching or chattering may occur. One way to resolve chattering is to consider weighted error history as well as the current error (Narendra and Mukhopadhyay, 1997),

$$J_i(t) = \alpha e_i^2(t) + \beta \sum_{j=1}^t \lambda^{(t-j)} e_i^2(j) \quad (2.79)$$

where  $e_i$  is the error of the  $i$ th model,  $\alpha$  and  $\beta$  are tuning parameters accounting for the instantaneous and weighted average residue, respectively. The problem of applying this strategy for multiple Laguerre models is that orthogonal LS does not explicitly calculate  $e_i(t)$ . Rather, the upper triangular matrix  $R$  contains the information of averaged error, which is equivalent to the second term of (2.79) combined

with forgetting factor. The computation of explicit  $e_i(t)$  will require additional computational burden. Instead, we are directed to hysteresis switching. In (Morse et al., 1992), it is proven that hysteresis switching converges in a finite time. Hysteresis switching can be represented as

$$\sigma(t) = \phi(\sigma(t-1), \delta(\hat{\sigma}(t))), \sigma(0) = i_0 \quad (2.80)$$

where

$$\phi(\sigma(t-1), \delta(\hat{\sigma}(t))) = \begin{cases} \sigma(t-1) & \text{if } \delta(\hat{\sigma}(t)) < \delta(\rho(t)) + h \\ \rho(t) & \text{if } \delta(\hat{\sigma}(t)) \geq \delta(\rho(t)) + h \end{cases} \quad (2.81)$$

Here,  $\sigma(t)$  is the index of selected model at  $t$ ,  $\hat{\sigma}(t)$  is the tentative index of the selected model and is  $\hat{\sigma}(t) = \sigma(t-1)$ ,  $\rho(t)$  is the index of the best local model at  $t$ ,  $\delta(t)$  is a value function of each local model such as residue. Hence,  $\delta(\hat{\sigma}(t))$  can be considered as the residue of the currently selected model while  $\delta(\rho(t))$  is the smallest residue.  $h$  is the threshold value.

The adaptive multiple Laguerre model identification is given as follows.

**Design parameters** the variables for irregular grid generation in Section 2.5.1,  $n$  the order of Laguerre models,  $n_m$  the maximum number of concurrent local Laguerre models that can exist at the same time,  $a_\lambda$  and  $b_\lambda$  for adaptive forgetting,  $y_{\lambda u}$  the upper bound of adaptation variable for adaptive forgetting,  $h$  the threshold value for hysteresis switching

**step 1** Generate  $m$  multiple Laguerre models using the algorithm from Sec. 2.5.1

**step 2** initialize  $\sigma(0)$  in (2.80)

**step 3** compute Laguerre states of all the local models by (2.73)

**step 4** update forgetting factor  $\lambda(t)$  by (2.76) and (2.78)

**step 5** time update by augmenting  $R(t-1)$  with  $x(t)$  and  $y(t)$  as

$$R(t) = \begin{bmatrix} R(t-1) \\ [x^T(t), y(t)] \end{bmatrix} \quad (2.82)$$

and take QR decomposition of  $R(t)$

**step 6** compute the error of the tentatively current model, that is,  $\delta(\hat{\sigma}(t))$  in (2.81) by (2.68)

**step 7** compute  $\delta(\rho(t))$  by permutating  $R(t)$  and compute residue by (2.68)

**step 8** decide the proper model and update  $\sigma(t)$  by hysteresis switching

**step 9** update the parameters  $\theta(t)$  and residue for the selected model  $\delta(\sigma(t))$  from  $R(t)$ , by solving orthogonal least squares

**step 10** go to step 3

## 2.6 Example

We consider a rapidly switching linear system described as the following.

$$G(z) = \begin{cases} \frac{1}{(z-0.7)(z-0.3)} & 0 \leq t < 50 \\ \frac{1}{(z+0.5)(z-0.3)} & 50 \leq t < 100 \\ \frac{1}{(z-0.3)(z-0.9)} & 100 \leq t < 150 \end{cases} \quad (2.83)$$

Generate normal random input  $u$  with  $N(0, 1)$ . Set the Laguerre order  $n = 3$  and the maximum concurrent number of models  $n_m = 2$ . 8 Laguerre models are generated when the design variables are chosen to be  $R = 0.3$ ,  $\beta = 0.3$  and  $\gamma = 10^{-3}$ . Initial  $y_\lambda(0) = 0.2$ .  $a_\lambda = 0.4$  and  $b_\lambda = 1$ . Threshold variable  $h = 10^{-3}$ . Initially,  $\sigma(0) = 1$ , that is, initial model is selected to be the first one. The performance of resulting multiple Laguerre model is shown in Fig. 2.3 compared with adaptive Laguerre model with a fixed pole at 0.3. As can be seen, the multiple Laguerre model has improved response.

## 2.7 Discussion

In this chapter, we developed adaptive multiple Laguerre filters to identify rapidly changing linear dynamics. The use of orthogonal least squares in combination with

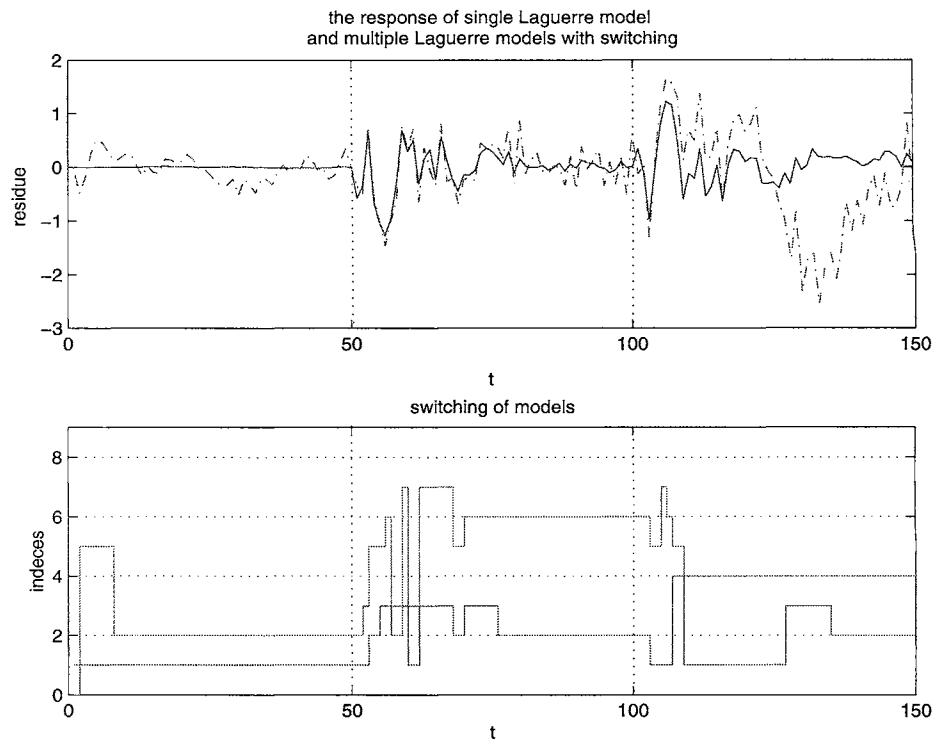


Figure 2.3: Adaptive multiple Laguerre model for rapidly switching system identification: (top figure) dotted line represents the error of single Laguerre model and solid line represents the residue of multiple Laguerre model. (bottom figure) the indices of multiple Laguerre models



multiple Laguerre models that have optimally representing individual regions resulted in improved representation of linear stable systems with reduced model complexity. The question at this moment would probably be ‘what is from now?’

The stability condition of linear system can certainly be the limiting factor to identify general nonlinear systems. Stochastic effects to the identification algorithm has to be analyzed. However, most of all, the adaptive character of the algorithm is the most deciding factor to limit the application of the model to control problems. One possibility with control can be predictive type controller, which utilizes the prediction capability of the model and obtains controller by solving optimization problem to reduce the set point error. The problem of this application of the model is the lack of analytical guarantee of stability and performance. The problem is raised from the asymptotic nature of optimality of Laguerre models while recursive least squares is with only finite number of data. The data for recursive least squares are finite because of forgetting factor, which is used to adapt to changing dynamics. Even though the proof of asymptotic convergence of Laguerre model to a single linear system is possible, it is far from practical use of the model for nonlinear control.

For this reason, we redirect our focus of research from online extension of linear system identification for control to off-line system identification for multimodel based control. We expect that the significance of off-line modelling must be great in order to support reliable controller design. Also, off-line identification is a prior step to online identification since online identification is usually only a recursive version of off-line identification with more difficult conditions to meet.

## Part II

# Study on the local model for multimodel-based nonlinear system identification

# Chapter 3

## On the local interpretation of Takagi-Sugeno fuzzy models from a dynamical systems point of view

### 3.1 Introduction

The Takagi-Sugeno (TS) fuzzy model was initially proposed as a method for static function approximation by decomposing input space by premise and representing the consequence by constant-affine input-output equations (Takagi and Sugeno, 1985). The output  $y$  for the input vector  $x = [x_1, x_2, \dots, x_n]^T$  with input dimension  $n$  is represented by  $m$  local models as follows:

$$\begin{aligned} y &= \frac{\sum_{i=1}^m \rho_i(x)(a_1^i x_1 + \dots + a_n^i x_n + a_{n+1}^i)}{\sum_{j=1}^m \rho_j(x)}, \\ &= \sum_{i=1}^m w_i(x) y_i(x), \end{aligned}$$

and locally by

$$y_i(x) = a_1^i x_1 + \dots + a_n^i x_n + a_{n+1}^i,$$

where the fuzzy membership function for the  $i$ th model,  $\rho_i(\cdot)$ , is a tensor product as  $\rho_i(x) = \rho_{i1}(x_1) \times \rho_{i2}(x_2) \times \dots \times \rho_{in}(x_n)$ .  $w_i(x)$  is a normalized fuzzy membership function as  $w_i(x) = \rho_i(x) / \sum_{j=1}^m \rho_j(x)$ .

Since this model is linear-in-the-parameters, once fuzzy membership functions are fixed, the coefficients,  $a_l^i, i = 1, \dots, m, l = 1, \dots, n + 1$  can be efficiently estimated by

linear least squares (LS) as

$$\begin{aligned}
y(k) &= \sum_{i=1}^m a_1^i w_i x_1(k) + a_2^i w_i x_2(k) + \dots + a_n^i w_i x_n(k) + a_{n+1}^i w_i, \\
&= [w_1[x_1(k), \dots, x_n(k), 1], \dots, w_m[x_1(k), \dots, x_n(k), 1]] \cdot \\
&\quad [a_1^1, \dots, a_{n+1}^1, \dots, a_1^m, \dots, a_{n+1}^m]^T, \\
&= z(k)^T \beta, \quad k = 1, \dots, N
\end{aligned}$$

and linear LS estimation is

$$\beta = (Z^T Z)^{-1} Z^T Y,$$

where  $Z = [z(1)^T, z(2)^T, \dots, z(N)^T]^T$  and  $Y = [y(1), y(2), \dots, y(N)]^T$ .  $N$  is the number of data available. In the original formation, the premise is chosen by reasoning or tree structure (hierarchical) search by minimizing an error criterion.

This structure is later adopted in the so called Multiple-Model (MM) frameworks for approximation of nonlinear dynamical systems and shown to be optimal in the sense of minimizing the criterion

$$J = \sum_{i=1}^m \int_{x \in X} \|\hat{f}(x) - \hat{f}_i(x)\|_2^2 \rho_i(x) dx,$$

where  $\hat{f}(\cdot)$  is a global model, and  $\hat{f}_i(\cdot)$  is the  $i$ th local model (Johansen, 1994). For the approximation of dynamical systems instead of static functions, input space  $X$  is replaced with regressor space  $\Psi$ . In general, we choose  $\varphi(t) = [y(t-1), y(t-2), \dots, y(t-ny), u(t), u(t-1), \dots, u(t-nu)]^T \in \Psi$ , where  $t$  is a time index with input  $u(t)$  and output  $y(t)$ . Then the *constant-affine recursive i/o equation* is as

$$y(t) = \varphi^T(t)\theta + d_0, \tag{3.1}$$

where  $d_0$  is a constant.

The adoption of the TS fuzzy model structure for dynamical systems is justified by considering the nonlinear systems as the Nonlinear ARMAX (NARMAX) model representation and linearization of it to obtain the TS fuzzy model structure (Johansen,

1994). Models by multimodels system identification using the TS fuzzy model structure for system analysis and controller design is, however, problematic.

One problem is in the identification process. The identification of model parameters based on global criterion is suggested by (Takagi and Sugeno, 1985) as

$$\begin{aligned} J &= \sum_{t=0}^{N-1} (y(t) - \hat{y}(t))^2, \\ &= \sum_{t=0}^{N-1} \left( y(t) - \sum_{i=1}^m w_i(t) \hat{y}_i(t) \right)^2, \end{aligned}$$

which can be rewritten as

$$J = \sum_{t=0}^{N-1} \left( \sum_{i=1}^m w_i(t) (y(t) - \hat{y}_i(t)) \right)^2, \quad (3.2)$$

because  $\sum_{i=1}^m w_i(t) = 1$  for all  $t$ . The estimation based on this error criterion does not necessarily produce local models representing local characteristics. In (Murray-Smith and Johansen, 1997, see chap. 7 and chap. 8), this phenomena is shown by simulations. However, it is not difficult to see the problem of local dynamics identification. Consider a linear-in-the-parameters local model as  $\hat{y}_i(t) = \varphi(t)^T \theta^i$  with a global objective function (3.2). The least square estimator is derived from the condition

$$\begin{aligned} \frac{\partial J}{\partial \theta^k} &= \sum_{t=0}^{N-1} 2 \left( \sum_{i=1}^m w_i(t) (y(t) - \varphi(t)^T \theta^i) \right) \frac{\partial}{\partial \theta^k} \left( \sum_{i=1}^m w_i(t) (y(t) - \varphi(t)^T \theta^i) \right), \\ &= -2 \sum_{t=0}^{N-1} \sum_{i=1}^m w_i(t) w_k(t) (y(t) - \varphi(t)^T \theta^i) \varphi(t) = 0, \quad k = 1, 2, \dots, m. \end{aligned} \quad (3.3)$$

Hence the parameters of local models are coupled, and the LS estimator is to minimize the global prediction error.

In contrast, the local objective criterion is given as

$$\begin{aligned} J_i &= \sum_{t=0}^{N-1} (w_i(t) y(t) - w_i(t) \varphi(t)^T \theta^i)^2, \\ &= \sum_{t=0}^{N-1} w_i^2(t) (y(t) - \varphi(t)^T \theta^i)^2, \end{aligned} \quad (3.4)$$

and the summation of the local criteria is given as

$$J = \sum_{i=1}^m J_i. \quad (3.5)$$

The LS estimation of (3.5) is given as

$$\frac{\partial J}{\partial \theta^k} = -2 \sum_{t=0}^{N-1} w_k^2(t) (y(t) - \varphi^T(t) \theta^k) \varphi(t) = 0, \quad k = 1, 2, \dots, m, \quad (3.6)$$

which is simply weighted least square estimator. By choosing  $w_k(t)$  to localize the data, local dynamics can be identified. We can see that the parameter estimate using global error criterion (3.3) is same as the one with local criterion (3.6) if  $w_i(t)$  is orthogonal. This is the adverse effect of interpolation or overlapping weighting functions on local model identification. As is clear from (3.3), the local models estimated by minimizing the global error criterion cannot represent local dynamics.

Another problem, and the subject of this Chapter, is in the interpretation of the model in terms of dynamical systems. The additional offset term in the constant-affine local model is a reasonable choice for function approximation, however, the dynamical interpretation of it is not clear. There has been some effort devoted to this problem (Johansen, 2000; Shorten et al., 1999). The main approach is to consider nonlinear state space representation and interpret the TS fuzzy model as a linearized system. That is, given a nonlinear dynamic system as

$$\dot{x}(t) = f(x(t), u(t)), \quad (3.7)$$

$$y(t) = g(x(t), u(t)), \quad (3.8)$$

dynamic linearization (linearization along a trajectory) using Taylor series is given as

$$\begin{aligned} \dot{x}_L(t) &= f(x_0(t), u_0(t)) + \left. \frac{\partial f}{\partial x} \right|_0 (x_L(t) - x_0(t)) \\ &\quad + \left. \frac{\partial f}{\partial u} \right|_0 (u(t) - u_0(t)), \\ &= \left. \frac{\partial f}{\partial x} \right|_0 x_L(t) + \left. \frac{\partial f}{\partial u} \right|_0 u(t) \\ &\quad + f(x_0(t), u_0(t)) - \left. \frac{\partial f}{\partial x} \right|_0 x_0(t) - \left. \frac{\partial f}{\partial u} \right|_0 u_0(t). \end{aligned}$$

In result,

$$\dot{x}_L(t) = A(x_0, u_0)x_L(t) + B(x_0, u_0)u(t) + d_0(x_0, u_0), \quad (3.9)$$

where  $x_L(t)$  is a state vector of the linearized system, and  $[x_0^T, u_0^T]^T$  is the trajectory satisfying  $\dot{x}_0(t) = f(x_0(t), u_0(t))$ . Equation (3.9) is similar to constant-affine equation (3.1) but not exact since (3.1) is a i/o equation while (3.9) is a state equation. Therefore, there is a gap between this approach of interpretation and TS fuzzy models. In addition, the analysis of linear time varying systems is different from linear time invariant systems. In (Johansen, 2000; Shorten et al., 1999), state information is assumed to be available, however, it is very restrictive to make such an assumption. In most system identification problems, only the i/o information is available. Therefore, interpretation of dynamical systems must be pursued in i/o form.

In spite of the fuzziness in the interpretation of TS fuzzy models, TS fuzzy models have been popularly used for model based controller design by transforming the constant-affine i/o model (or linear i/o model) into state-space form by state realization (Kiriakidis et al., 1998; Cao et al., 1995). However, it is reasonable to understand what the model indeed represents ahead of the model based controller design since this type of controllers is bound to be limited by the quality of the model.

The main goal of this Chapter is to demonstrate the *limitations of TS fuzzy models for the local representation of dynamical systems from model structures point of view*. We treat dynamical systems as nonlinear state space representation, and local dynamics are obtained by linearization. A recursive i/o equation is derived by transforming this local dynamics. The resulting recursive i/o equation is turned out inexact with local TS fuzzy models. However, the difference is shown to be small in terms of i/o behaviors as far as the regression vector is properly chosen. A more serious problem surfaces in state space realization of TS fuzzy models. It is shown that the local state space *cannot* be recovered from TS fuzzy models; i.e. local state space is not identifiable by TS fuzzy models. In Section 3.2, dynamical systems in

state space are compared with TS fuzzy models, and the quality of TS fuzzy models for dynamic system identification is analyzed. A numerical example is presented in Section 3.3 to illustrate the concepts of this Chapter.

## 3.2 The relation between TS fuzzy models and dynamical systems

Consider the continuous time dynamical systems represented by (3.7) and (3.8). Our interest of dynamical systems in this representation is motivated by rich theories in controller design and system analysis in state space for the possible use of the identified models. By proper sampling, this system can be represented in discrete state space as

$$\begin{aligned}x(t+1) &= f_d(x(t), u(t)), \\y(t) &= g_d(x(t), u(t)),\end{aligned}$$

where  $t$  is in  $\mathbb{N}_0 = \{0, 1, 2, \dots\}$  rather than nonnegative real as in continuous time. Linearization of the discrete time nonlinear systems by Taylor series at an arbitrary point  $(x_0, u_0)$  gives

$$x_L(t+1) = A(x_0, u_0)x_L(t) + B(x_0, u_0)u(t) + F, \quad (3.10)$$

$$y(t) = C(x_0, u_0)x_L(t) + D(x_0, u_0)u(t) + G, \quad (3.11)$$

where  $x_L$  is the state of the linearized system, and  $F$  and  $G$  are offset terms caused by nonzero or off equilibriums. We assume that the linearized dynamics without the offset terms,  $F$  and  $G$ , is controllable as well as observable. Hereinafter,  $x$  instead of  $x_L$  will be used by ignoring the high order terms in the linearized dynamics.

It is tempting to relate (3.10) and (3.11) with local TS fuzzy models (3.1), however, the direct relation between these equations and TS fuzzy models cannot be established and will be shown in the following. Before that, we need to clarify the definition of the equivalence between models.



**Definition 3.1 (Model equivalence)** Two systems  $y_1(t) = f_1(u(t))$  and  $y_2(t) = f_2(u(t))$  are equivalent if and only if  $y_1(t) = y_2(t)$  for all  $t \in \mathbb{N}_0$  with zero initial conditions.

In the above definition, zero initial conditions are necessary since the same dynamical systems can have different outputs depending on the initial conditions. Zero initial conditions for systems in state space implies  $x(0) = 0$  while  $\varphi(t) = 0, t \leq 0$  for regression type representations. The definition of equivalence is given since infinitely many dynamic systems can have the same i/o data pairs (Hammer, 1984). One simple example is that the dynamic system  $x(t+1) = f(x(t), u(t))$  is equal to  $x(t+1) = f(f(x(t-1), u(t-1)), u(t))$ . In general, we want minimal i/o equations.

**Proposition 3.1 (Non-equivalence of local dynamics and affine regression model)** Suppose that  $C \text{adj}(zI - A)F + G \det(zI - A)$  is not a constant. Then the system represented by (3.10) and (3.11) is not equivalent to constant-affine time invariant recursive i/o equation (3.1).

**Proof:** We prove it by transforming the state space representation into a recursive i/o form and compare it with (3.1).

Take the  $z$ -transformation of (3.10), we obtain

$$(zI - A)X(z) = BU(z) + F \frac{z}{z-1}.$$

Then, with  $z$ -transformation of equation (3.11), the i/o equation in  $z$ -domain is given as

$$\begin{aligned} Y(z) &= (C(zI - A)^{-1}B + D)U(z) \\ &\quad + (C(zI - A)^{-1}F + G) \frac{z}{z-1}. \end{aligned}$$

To realize a recursive i/o equation, multiply both sides with  $z^{-n}M(z)$  then

$$\begin{aligned} z^{-n}M(z)Y(z) &= z^{-n}M(z) (C(zI - A)^{-1}B + D)U(z) \\ &\quad + z^{-n}M(z) (C(zI - A)^{-1}F + G) \frac{z}{z-1}, \end{aligned}$$

where  $M(z)$  is a polynomial function in  $z$  and  $n$  is the order of  $M(z)$ .  $z^{-n}$  is an additional multiplication factor to make  $y(t)$  the most forwarded term in time. For the above equation to be constant-affine recursive i/o form,

$$z^{-n}M(z) (C(zI - A)^{-1}F + G) \frac{z}{z-1} = d_0 \frac{z}{z-1}, \quad (3.12)$$

where  $d_0$  is a constant. The above equation is satisfied if and only if

$$\begin{aligned} M(z) &= \frac{d_0 z^n}{C(zI - A)^{-1}F + G}, \\ &= \frac{d_0 z^n \det(zI - A)}{C \operatorname{adj}(zI - A)F + G \det(zI - A)}, \end{aligned}$$

which is not a polynomial function unless the denominator is a nonzero constant. □

The proof of the proposition shows that the offset term of the recursive i/o equation of linearized dynamics is indeed a function of time instead of a constant. A typical choice of  $M(z)$  is  $\lambda(z)p(z)$ , where  $\lambda(z)$  is the characteristic equation and  $p(z)$  is an arbitrary polynomial function. From (3.12), we can see that left hand side of the equation is actually a time varying term, not a constant. Since the term is a polynomial of  $z^{-1}$  with finite order divided by  $z/(z-1) = 1 - z^{-1}$ , it only has finite duration in addition to a constant. If  $M(z)$  has  $z-1$  as a factor, this term can last at most  $n$  steps.

Since the proposition shows that the local TS fuzzy model is not an exact representation of local linear dynamics, it is interesting to see how TS fuzzy models behave in system identification of the local dynamics. For this purpose, consider the linearized system as

$$\begin{aligned} y(t) &= a_1 y(t-1) + \dots + a_{ny} y(t-ny) \\ &\quad + b_0 u(t) + \dots + b_{nu} u(t-nu) + d_0 + d(t), \end{aligned} \quad (3.13)$$

$$= \gamma^T(t) \theta_e, \quad (3.14)$$

where  $\theta_e = [a_1, \dots, a_{ny}, b_0, \dots, b_{nu}, d_0, 1]^T = [\theta^T, d_0, 1]^T$  and  $\gamma(t) = [y(t-1), \dots, y(t-ny), u(t), \dots, u(t-nu), 1, d(t)]^T = [\varphi^T(t), 1, d(t)]^T$ .  $d_0$  is a constant, and  $d(t)$  is a time varying offset term with finite duration. In the following Property 3.1, the estimation error is shown in case that the system is approximated by the constant-affine i/o model as (3.1). Another popular local structure for the TS fuzzy model is a linear model without the constant offset term. In Property 3.2, the behavior of linear models for the approximation of local dynamics is shown.

**Property 3.1 (Local model error of affine recursive i/o model)** Suppose the parameters are estimated by LS and  $\Phi^T(NI - \mathbf{1}\mathbf{1}^T)\Phi$  is nonsingular, then

$$\hat{Y} - Y = (NI - \mathbf{1}\mathbf{1}^T) (\Phi(\Phi^T(NI - \mathbf{1}\mathbf{1}^T)\Phi)^{-1}\Phi^T(NI - \mathbf{1}\mathbf{1}^T) - I) \frac{D_N}{N}. \quad (3.15)$$

where  $\Phi = [\varphi(1), \varphi(2), \dots, \varphi(N)]^T$ ,  $D_N = [d(1), d(2), \dots, d(N)]^T$ ,  $\mathbf{1}$  is  $N \times 1$  vector with 1's as the elements of the vector.  $\hat{Y}$  denotes the estimated model output vector.

**Proof:** In a matrix-vector form, the estimate

$$Y = [ \Phi \mid \mathbf{1} \mid D_N ] \begin{bmatrix} \theta \\ d_0 \\ \mathbf{1} \end{bmatrix} = [ \Phi \mid \mathbf{1} ] \begin{bmatrix} \theta \\ d_0 \end{bmatrix} + D_N \quad (3.16)$$

and

$$\hat{Y} = [ \Phi \mid \mathbf{1} ] \begin{bmatrix} \hat{\theta} \\ \hat{d}_0 \end{bmatrix}. \quad (3.17)$$

Then

$$\hat{Y} - Y = \Phi_e(\hat{\theta}_e - \theta_e) - D_N = \Phi_e\tilde{\theta}_e - D_N \quad (3.18)$$

where  $\Phi_e = [\Phi \mid \mathbf{1}]$ ,  $\theta_e = [\theta^T \ d_0]^T$  and  $\tilde{\theta}_e = \hat{\theta}_e - \theta_e$ .

Normal equation for the LS estimate is

$$[ \Phi \mid \mathbf{1} ]^T [ \Phi \mid \mathbf{1} ] \begin{bmatrix} \hat{\theta} \\ \hat{d}_0 \end{bmatrix} = [ \Phi \mid \mathbf{1} ]^T Y.$$

Replace  $Y$  in the normal equation with (3.16), the normal equation in block matrices

is

$$\begin{bmatrix} \Phi^T\Phi & \Phi^T\mathbf{1} \\ \mathbf{1}^T\Phi & \mathbf{1}^T\mathbf{1} \end{bmatrix} \begin{bmatrix} \hat{\theta} \\ \hat{d}_0 \end{bmatrix} = \begin{bmatrix} \Phi^T\Phi & \Phi^T\mathbf{1} & \Phi^TD_N \\ \mathbf{1}^T\Phi & \mathbf{1}^T\mathbf{1} & \mathbf{1}^TD_N \end{bmatrix} \begin{bmatrix} \theta \\ d_0 \\ \mathbf{1} \end{bmatrix}.$$

Define  $\tilde{\theta} = \hat{\theta} - \theta$  and  $\tilde{d}_0 = \hat{d}_0 - d_0$ ,

$$\left[ \begin{array}{c|c} \Phi^T \Phi & \Phi^T \mathbf{1} \\ \hline \mathbf{1}^T \Phi & \mathbf{1}^T \mathbf{1} \end{array} \right] \begin{bmatrix} \tilde{\theta} \\ \tilde{d}_0 \end{bmatrix} = \begin{bmatrix} \Phi^T D_N \\ \mathbf{1}^T D_N \end{bmatrix}$$

Instead of using the matrix inversion lemma, we can directly obtain

$$\Phi^T \Phi \tilde{\theta} + \Phi^T \mathbf{1} \tilde{d}_0 = \Phi^T D_N, \quad (3.19)$$

$$\mathbf{1}^T \Phi \tilde{\theta} + \tilde{d}_0 N = \mathbf{1}^T D_N. \quad (3.20)$$

From (3.20),

$$\tilde{d}_0 = (\mathbf{1}^T D_N - \mathbf{1}^T \Phi \tilde{\theta}) / N,$$

then with (3.19),

$$\Phi^T \Phi \tilde{\theta} + \Phi^T \mathbf{1} (\mathbf{1}^T D_N - \mathbf{1}^T \Phi \tilde{\theta}) / N = \Phi^T D_N.$$

Solve for  $\tilde{\theta}$

$$\begin{aligned} \tilde{\theta} &= (\Phi^T \Phi - \Phi^T \mathbf{1} \mathbf{1}^T \Phi / N)^{-1} \Phi^T (I - \mathbf{1} \mathbf{1}^T / N) D_N, \\ &= (\Phi^T (NI - \mathbf{1} \mathbf{1}^T) \Phi)^{-1} \Phi^T (NI - \mathbf{1} \mathbf{1}^T) D_N. \end{aligned} \quad (3.21)$$

Use this equation and (3.20), we get

$$\tilde{d}_0 = \mathbf{1}^T (I - \Phi (\Phi^T (NI - \mathbf{1} \mathbf{1}^T) \Phi)^{-1} \Phi^T (NI - \mathbf{1} \mathbf{1}^T)) D_N / N \quad (3.22)$$

Replace  $\tilde{\theta}_e$  in (3.18) with (3.21) and (3.22), and we obtain

$$\hat{Y} - Y = (NI - \mathbf{1} \mathbf{1}^T) (\Phi (\Phi^T (NI - \mathbf{1} \mathbf{1}^T) \Phi)^{-1} \Phi^T (NI - \mathbf{1} \mathbf{1}^T) - I) \frac{D_N}{N}.$$

□

**Remark 3.1** *From the Property 3.1, we can see that residue of the constant affine i/o model can be kept small with proper regression vector since  $D_N$  has only a few nonzero terms.*

The number of training data  $N$  does not have influence on the residues. This can be easily seen by considering the singular value decomposition of  $NI - \mathbf{1}\mathbf{1}^T$ . Since  $N - 1$  singular values of the matrix is  $N$  and the last singular value is 0, the singular value decomposition is

$$\begin{aligned} NI - \mathbf{1}\mathbf{1}^T &= [X_1 \ x] \begin{bmatrix} NI & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X_1^T \\ x^T \end{bmatrix} \\ &= NX_1X_1^T. \end{aligned} \quad (3.23)$$

Therefore,

$$\begin{aligned} \hat{Y} - Y &= NX_1X_1^T (\Phi(N\Phi^T X_1X_1^T \Phi)^{-1} \Phi^T NX_1X_1^T - I) \frac{D_N}{N} \\ &= X_1X_1^T (\Phi(\Phi^T X_1X_1^T \Phi)^{-1} \Phi^T X_1X_1^T - I) D_N. \end{aligned} \quad (3.24)$$

**Property 3.2 (Local model error of linear recursive i/o model)** Suppose the same assumptions that was made in Property 3.1. For a linear model in regression form as

$$\hat{y}(t) = \varphi^T(t)\hat{\theta},$$

then the residue vector is

$$\hat{Y} - Y = (\Phi(\Phi^T \Phi)^{-1} \Phi^T - I)(D_N + \mathbf{1}d_0),$$

where all the notations are followed from Property 3.1.

**Proof:** The output vector can be rewritten as

$$Y = \Phi\theta + (D_N + \mathbf{1}d_0)$$

while the model output vector is in matrix-vector form as

$$\hat{Y} = \Phi\hat{\theta}.$$

Using the solution of normal equation and the matrix-vector form of outputs,

$$\begin{aligned} \hat{\theta} &= (\Phi^T \Phi)^{-1} \Phi^T Y, \\ &= (\Phi^T \Phi)^{-1} \Phi^T (\Phi\theta + D_N + \mathbf{1}d_0), \\ &= \theta + (\Phi^T \Phi)^{-1} \Phi^T (D_N + \mathbf{1}d_0). \end{aligned}$$

Hence,

$$\begin{aligned}\hat{Y} - Y &= \Phi(\hat{\theta} - \theta) - (D_N + \mathbf{1}d_0) \\ &= (\Phi(\Phi^T\Phi)^{-1}\Phi^T - I)(D_N + \mathbf{1}d_0),\end{aligned}$$

□

**Remark 3.2** *Property 3.2 shows that residue for the linear local models does not converge to zero, that is, the estimation is biased. This shows that linear TS fuzzy local model cannot even approximate the local i/o behavior of the system and the use of this local model must be cautiously made.*

**Remark 3.3** *It is worthy of mentioning that TS fuzzy models with linear local models are originated from continuous time domain with nonlinear model available (Tanaka et al., 1998). It is certainly possible to obtain accurate linear local models by formulating model nonlinearities as fuzzy premises. However, as shown in the property, it is dangerous to extend the approach to discrete-time domain that involves system identification. It is also difficult to define accurate premise to validate linear local models without the knowledge of the system.*

It is also interesting to see how the constant-affine model is transformed to state space. One popular approach is to completely ignore the offset term, that is, consider a local model as the one in Property 3.2 (Cao et al., 1995). The state realization of this linear i/o equation is trivial, however, the interpretation of the model is vague.

Constant-affine model (3.1) may be transformed into the form similar to (3.10) and (3.11) by choosing  $F$  and  $G$  to satisfy the condition (3.12). A simple realization can be obtained by first getting state realization of the linear part and solving the above condition to determine  $F$  and  $G$ . In (Kiriakidis et al., 1998), observable canonical form is used for the realization and  $G = d$  and  $F = [a_1, a_2, \dots, a_{ny}]^T d$  are selected,

where  $a_i$  are from (3.13). However, this choice of offset terms is rather arbitrary and the physical interpretation is misleading as well.

Our investigation of the relationship between the constant-affine state model and the constant-affine i/o model, indeed, reveals that a model with (3.10) and (3.11) is not identifiable by (3.1). Before presenting the result, we extend the definition of model identifiability in (Ljung, 1999) by combining with the concept of model equivalence in Definition 3.1. The definition focuses on one-to-one mapping between parameters of two different model structures.

**Definition 3.2 (Extended identifiability between different model structures)**

A model structure  $M_1$  is identifiable at  $\theta^1$  by  $M_2$  at  $\theta^2$  if and only if there is a function  $f(\cdot)$ , such that  $\theta^1 = f^{-1}(\theta^2)$  and  $M_1(\theta^1)$  and  $M_2(\theta^2)$  are equivalent according to Definition 3.1.

**Proposition 3.2 (Unidentifiability of local dynamics by affine recursive i/o model)** Suppose the linearized system, (3.10) and (3.11), is locally controllable and observable, that is,  $(A, C)$  is observable and  $(A, B)$  is controllable. Then, the local dynamics is not identifiable by constant-affine recursive i/o models as (3.1).

**Proof:** We prove by induction. Consider the constant-affine state equation represented by (3.10) and (3.11). First, consider a 2nd order system. Since the local dynamics is assumed to be controllable and observable, it is convenient to convert the model into canonical form whose system matrices are defined as

$$\begin{aligned} x(t+1) &= \begin{bmatrix} a_1 & 1 \\ a_2 & 0 \end{bmatrix} x(t) \\ &+ \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(t) + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \\ y(t) &= [1 \ 0] x(t) + du(t) + g. \end{aligned}$$

Realize recursive i/o form by utilizing  $z$ -transform and we obtain

$$(1 - a_1z^{-1} - a_2z^{-2})Y(z) = (d + (b_1 - a_1d)z^{-1} + (b_2 - a_2d)z^{-2})U(z) \\ + (g + (f_1 - a_1g)z^{-1} + (f_2 - a_2g)z^{-2})\frac{1}{1 - z^{-1}}.$$

The time domain realization of the affine term is

$$gu_s(t) + (f_1 - a_1g)u_s(t - 1) + (f_2 - a_2g)u_s(t - 2)$$

and can be rewritten as

$$(g + (f_1 - a_1g) + (f_2 - a_2g))u_s(t) - (f_1 - a_1g)\delta(t) - (f_2 - a_2g)\delta(t - 1)$$

where  $u_s(t)$  is the unit step function, and  $\delta(t)$  is the unit impulse function. Since the time varying offset terms last finitely, the LS estimation results in

$$\hat{d}_0 \approx (g + (f_1 - a_1g) + (f_2 - a_2g)). \quad (3.25)$$

Then the estimated output becomes

$$\hat{Y}(z) = \frac{b_1z^{-1} + b_2z^{-2}}{1 - a_1z^{-1} - a_2z^{-2}}U(z) \\ + \frac{\hat{d}_0}{(1 - a_1z^{-1} - a_2z^{-2})(1 - z^{-1})}. \quad (3.26)$$

Since (3.25) is an underdetermined equation, there is an infinite number of solutions for  $(f_1, f_2, g)$ . We generalize this result to a  $n$ th order system and we get

$$\hat{d}_0 = \sum_{i=1}^n f_i + (1 - \sum_{i=1}^n a_i)g,$$

and this proves that the constant-affine state equation is not identifiable by the constant-affine recursive i/o model.

□



**Remark 3.4** *The above proposition is rather intuitive since state equation has more parameters than i/o equation, and one-to-one mapping between these parameters cannot be established. We made the assumption of canonical system to show that the unidentifiability of the model is not caused by state matrices. The proposition is considered since TS fuzzy models have been used without discretion to obtain state space representation in literatures.*

To further investigate TS models used in fuzzy model based control literatures, consider the model suggested by (Kiriakidis et al., 1998) used for proving the stability of model-based controller. Consider the 2nd order system in the proof of Proposition 3.2. Since  $\hat{d} = (f_1 + f_2 + (1 - a_1 - a_2)g)$ , the selection of  $g = d$ ,  $f_1 = a_1d$  and  $f_2 = a_2d$  satisfies (3.25). Ignore the parametric estimation error and only consider the output error between the system and the model, then

$$\begin{aligned}\Delta Y(z) &= Y(z) - \hat{Y}(z), \\ &= \frac{z(gz^2 + (f_1 - a_1g)z + (f_2 - a_2g) - \hat{d})}{(z^2 - a_1z - a_2)(z - 1)}, \\ &= \frac{z(gz^2 + (f_1 - a_1g)z + (f_2 - a_2g)) - (f_1 + f_2 + (1 - a_1 - a_2)g)}{(z^2 - a_1z - a_2)(z - 1)}.\end{aligned}$$

Use the final value theorem

$$\begin{aligned}\lim_{t \rightarrow \infty} \Delta y(t) &= \lim_{z \rightarrow 1} (1 - z^{-1}) \Delta Y(z), \\ &= \frac{g + (f_1 - a_1g) + (f_2 - a_2g) - (f_1 + f_2 + (1 - a_1 - a_2)g)}{1 - a_1 - a_2} \\ &= 0.\end{aligned}$$

From the equation, we can see that the output error becomes zero even if the offset terms are incorrect. To see how states behave, take  $z$ -transform of the state equations of the 2nd order system. Similarly, only consider state error  $\Delta X(z) = X(z) - \hat{X}(z)$

caused by the incorrect affine term, which is given by

$$\Delta X(z) = \frac{z}{(z-1)(z^2 - a_1z - a_2)} \begin{bmatrix} z & 1 \\ a_2 & z - a_1 \end{bmatrix} \cdot \left( \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} - \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} (f_1 + f_2 + (1 - a_1 - a_2)g) \right).$$

Use the final value theorem

$$\begin{aligned} \lim_{t \rightarrow \infty} \Delta x(t) &= \lim_{z \rightarrow 1} (1 - z^{-1}) \Delta X(z), \\ &= \frac{1}{1 - a_1 - a_2} \begin{bmatrix} 1 & 1 \\ a_2 & 1 - a_1 \end{bmatrix} \cdot \\ &\quad \left( \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} - \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} (f_1 + f_2 + (1 - a_1 - a_2)g) \right), \\ &\neq 0 \end{aligned}$$

This example clearly demonstrates the misleading information of the unidentifiable model structure and suggests the need of a new system identification scheme to identify the system.

### 3.3 Example

In this section, we illustrate the identification of local dynamics by an example.

Consider a nonlinear system given by the following nonlinear state space equations

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= -10\sin(x_1(t)) - 0.5x_2(t) + u(t), \\ y(t) &= e^{-x_1(t)}. \end{aligned}$$

In general, a numerical integration method is used to discretize continuous time systems. Even though the Euler method is the simplest, it requires extremely small integration steps and dynamics of the system is not representative. We consider the modified Euler method (second order Runge-Kutta method) to discretize the system.

Then the system is represented in discrete state space by

$$\begin{aligned}
x_1(t+1) &= x_1(t) - 5T^2 \sin(x_1(t)) + Tx_2(t) \\
&\quad - 0.25T^2 x_2(t) + T^2/2u(t), \\
x_2(t+1) &= -5T \sin(x_1(t)) - 0.5Tx_2(t) + Tu(t), \\
&\quad - 5T \sin(x_1(t) + x_2(t)T) + 2.5T^2 \sin(x_1(t)) \\
&\quad + .125x_2(t)T^2 - .25T^2u(t) + x_2(t), \\
y(t) &= e^{-x_1(t)},
\end{aligned}$$

where  $T$  is the sampling period (integration step). Choose  $T = 0.1$ , and linearize the system at an arbitrary point  $(x_{10}, x_{20}, u_0) = (\pi/4, 0, 7)$ , and we obtain

$$\begin{aligned}
x(t+1) &= Ax(t) + Bu(t) + F \\
y(t) &= Cx(t) + Du(t) + G,
\end{aligned}$$

where  $x(t) = [x_1(t) \ x_2(t)]^T$ ,  $A = \begin{bmatrix} 0.965 & 0.0975 \\ -0.689 & 0.916 \end{bmatrix}$ ,  $B = \begin{bmatrix} 0.005 \\ 0.0975 \end{bmatrix}$ ,  $C = [ -0.456 \ 0 ]$ ,  $D = 0$ ,  $F = \begin{bmatrix} -0.00791 \\ -0.1483 \end{bmatrix}$  and  $G = 0.814$ .

In order to obtain a recursive i/o equation, take  $z$ -transform,

$$\begin{aligned}
Y(z) &= \frac{-2.28 \times 10^{-3}z - 2.25 \times 10^{-3}}{z^2 - 1.881z + 0.951}U(z) \\
&\quad + \frac{3.61 \times 10^{-3}z + 3.29 \times 10^{-3}}{z^2 - 1.881z + 0.951} \frac{z}{z-1} + \frac{z}{z-1}0.814.
\end{aligned}$$

Choose the multiplying polynomial to be  $M(z) = \det(zI - A)$  and multiply the above equation by  $z^{-2}M(z)$  and we get

$$\begin{aligned}
(1 - 1.881z^{-1} + 0.951z^{-2})Y(z) \\
&= (-2.28 \times 10^{-3}z^{-1} - 2.25 \times 10^{-3}z^{-2})U(z) \\
&\quad + \frac{0.0640}{1 - z^{-1}} - 0.775z^{-1} + 0.75.
\end{aligned}$$

Hence, the *recursive i/o equation* is

$$\begin{aligned} y(t) - 1.881y(t-1) + 0.951y(t-2) \\ = -2.28 \times 10^{-3}u(t-1) - 2.25 \times 10^{-3}u(t-2) \\ + 0.75\delta(t) - 0.775\delta(t-1) + 0.0640u_s(t). \end{aligned}$$

If we choose  $M(z) = \det(zI - A)(z-1)$ , then the i/o equation in  $z$ -domain is

$$\begin{aligned} (1 - 2.88z^{-1} + 2.83z^{-2} - 0.951z^{-3})Y(z) \\ = (-2.28 \times 10^{-3}z^{-1} + 3.36 \times 10^{-5}z^{-2} + 2.25 \times 10^{-3}z^{-3})U(z) \\ + 0.814 - 1.528z^{-1} + 0.777z^{-2}, \end{aligned}$$

and the *recursive i/o form* is

$$\begin{aligned} y(t) - 2.88y(t-1) + 2.83y(t-2) - 0.951y(t-3) \\ = -2.28 \times 10^{-3}u(t-1) + 3.36 \times 10^{-5}u(t-2) + 2.25 \times 10^{-3}u(t-3) \\ + 0.814\delta(t) - 1.528\delta(t-1) + 0.777\delta(t-2). \end{aligned}$$

Excite the system with Gaussian random input, with mean 7 and unit variance. The input is chosen such that the system is excited around the domain that we are interested in. Choose the regression vector to be

$$\varphi(t) = [y(t-1), y(t-2), u(t-1), u(t-2), 1]^T.$$

The estimated parameter is

$$\theta = [1.881, -0.951, -2.28 \times 10^{-3}, -2.25 \times 10^{-3}, 0.0640].$$

The estimate with a higher regression vector  $\varphi(t) = [y(t-1), y(t-2), y(t-3), u(t-1), u(t-2), u(t-3)]$  is given as

$$\theta = [2.88, -2.83, 0.0951, -2.28 \times 10^{-3}, 3.36 \times 10^{-5}, 2.25 \times 10^{-3}].$$

In this estimation,  $N = 1000$ . As expected, the estimate is quite accurate if the regression vector is properly chosen.

Now, we show how the state space representation, which is obtained from the TS model, behaves. From the estimated recursive i/o equation, we can obtain an observable canonical form with offset terms as

$$\begin{aligned} A &= \begin{bmatrix} 1.881 & 1 \\ -0.9511 & 0 \end{bmatrix} \\ B &= [-0.00228 \quad -0.002246]^T \\ C &= [1 \quad 0] \\ F &= [1.2033 \times 10^{-1} \quad -6.08 \times 10^{-2}]^T \\ G &= 0.06397. \end{aligned}$$

Using the similarity transformation,  $T = \begin{bmatrix} -2.19 & 0 \\ -20.6 & -22.5 \end{bmatrix}$ , compare the true offset terms with the estimated offset terms

$$\begin{aligned} \hat{F} &= T\tilde{F}, \\ &= [-2.64 \times 10^{-1} \quad -1.11]^T, \\ \hat{G} &= 6.397 \times 10^{-2}, \end{aligned}$$

which are different from the true offset terms. This bias causes the estimated states to drift from true states, while estimated outputs converge to the true output as shown in Fig. 3.1.

### 3.4 Discussion

In this chapter, we analyzed the significance of TS fuzzy models with recursive i/o model structure from dynamical systems point of view, which is the common forms in fuzzy identification and control. Our analysis shows that recursive i/o form is ambiguous in representing local dynamics and it is too arbitrary to obtain state space model from recursive i/o form. The problem is caused because algebraic condition

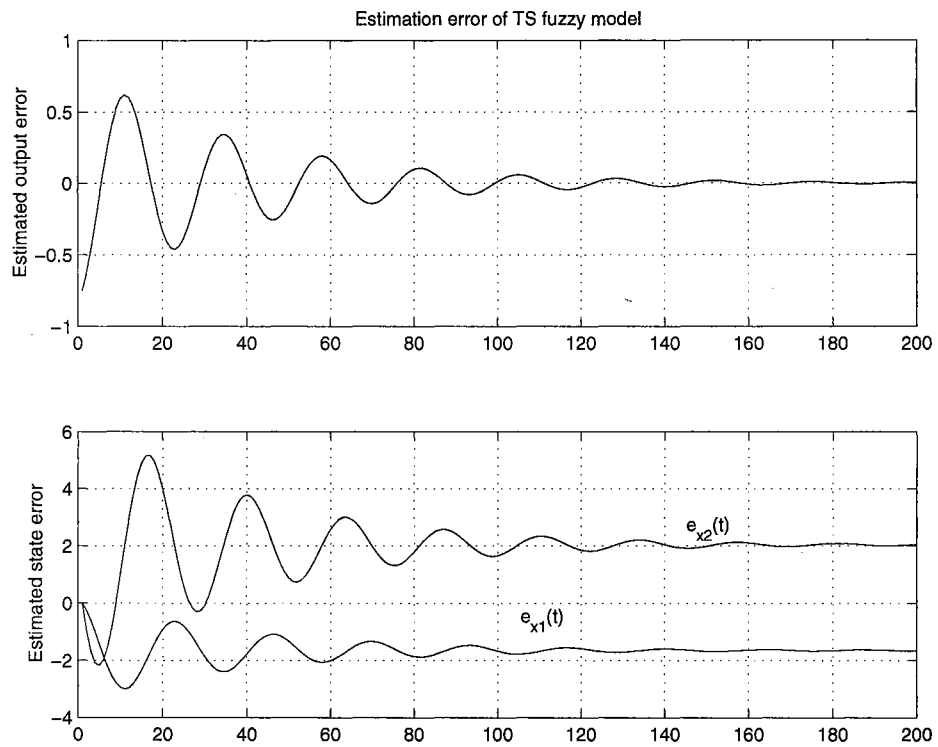


Figure 3.1: Error of the TS fuzzy model; the output error decays to zero (Top figure), while state errors drift (Bottom figure).

to obtain states bias is not satisfied. It is surprising that this problem has never been taken seriously. We suspect this ignorance of problem may be caused by the general attitude of system identification as ‘input-output mapping.’ We believe that considering system identification as only an i/o mapping problem can be dangerous for critical applications such as control and cannot justify its claim as the identification of dynamical systems. Because of this problem, we investigate different model structure and attempt to develop system identification methods in the next chapter.

# Chapter 4

## On the local identification of a Takagi-Sugeno fuzzy model in state space

### 4.1 Introduction

In Chapter 3, we pointed out the problem of TS fuzzy models in terms of the ambiguity of representing local dynamics. The argument is based on the assumption that ‘true system’ is in state space and the comparison of linearized state space form is used to point out the problems of TS fuzzy models in recursive i/o form. In this chapter, we further this argument by proposing Constant Affine State Space (CASS) model as a proper local model form because of its clear interpretation as local pseudo-linearized form. Also, we review some of the current development in multimodel based control, in order to support CASS model structure. Then, we investigate the CASS model structure for identification purpose. Unfortunately, CASS model structure is turned out to be unidentifiable in the presence of unknown initial states. We investigate the implication of unidentifiability of CASS model in various ways.

This chapter is organized as follows. In Section 4.2, we introduce CASS model and present issues to support this model structure for control. In Section 4.3, properties of CASS model are investigated from the perspective of system identification. In Section 4.4, we investigate the implication of unidentifiability of CASS model from



several different perspectives in estimation.

## 4.2 Constant-affine state space model for control

While linear system identification has been developed extensively in various model structures such as state space, recursive i/o, as well as transfer functions, and the relationships among them are well-established, nonlinear system identification is not the case. A popular approach in black-box nonlinear system identification is to replace the input space with regression space and treat nonlinear dynamical systems as a mapping from the regression space into output space (Sjöberg et al., 1995). Hence, the regression type identification is a mere extension of static function identification. The validation of the model is usually to compare the measured i/o data with the simulated or predicted outputs of the model. Since only finite data are available in practice, the validity of the model is very limited and it is hard to build enough confidence for real applications. For these problems, the application of the nonlinear model is quite limited to, e.g. prediction or simulation.

In addition to the confidence problem, the regression models have limitations in model based system analysis or controller design, since most of model based methods are based on state space models. To overcome the limitations of regression type models, some efforts were taken to derive a state space model from the regression model by nonlinear state realization (Sadegh, 1998). However, this equation based nonlinear state realization algorithm requires to find nonlinear mapping in order to obtain minimal state realization. Also, regression type models, such as nonlinear ARMAX, are only capable of representing nonlinear state space system locally (Srinivasan et al., 1994). On the other hand, recurrent neural networks with inner state dynamics have been proposed to emulate dynamical systems more closely (Jin and Gupta, 1995). However, the complex model structure of the recurrent neural networks hinders the model-based applications.

This motivates us to explore multimodel frameworks to cover global nonlinear systems with simpler local models. As pointed out in Chapter 3, the typical constant-affine recursive i/o model is not transparent for the interpretation of dynamical systems. Therefore, we are focused on local models in state space, especially, the constant-affine state space (CASS) model.

Constant-affine state space models naturally arise from the linearization of nonlinear state space models. Hence, this model structure can be utilized for the local representation of global nonlinear dynamics. We consider a SISO nonlinear system in state space as

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

$$y(t) = g(x(t), u(t)), \quad (4.2)$$

where  $x(t)$ ,  $u(t)$  and  $y(t)$  are state, input and output, respectively.  $t$  is the time index, and  $t \in \mathbb{N}_0$ , the set of nonnegative integers. Linearization of the above equations, (4.1) and (4.2), at an arbitrary point  $(x_0, u_0)$  results in

$$\begin{aligned} x(t+1) &= \nabla_x f|_0 (x(t) - x_0) \\ &\quad + \nabla_u f|_0 (u(t) - u_0) + f(x_0, u_0) \\ &= \nabla_x f|_0 x(t) + \nabla_u f|_0 u(t) \\ &\quad + f(x_0, u_0) - \nabla_x f|_0 x_0 - \nabla_u f|_0 u_0 \\ y(t) &= \nabla_x g|_0 (x(t) - x_0) \\ &\quad + \nabla_u g|_0 (u(t) - u_0) + g(x_0, u_0) \\ &= \nabla_x g|_0 x(t) + \nabla_u g|_0 u(t) \\ &\quad + g(x_0, u_0) - \nabla_x g|_0 x_0 - \nabla_u g|_0 u_0, \end{aligned}$$

where  $\nabla_x f|_0$  represents  $\frac{\partial f}{\partial x}$  at  $(x_0, u_0)$ . Collect the constant matrices and vectors, and

we obtain the CASS model as

$$x(t+1) = A(\theta)x(t) + B(\theta)u(t) + F(\theta), \quad (4.3)$$

$$y(t) = C(\theta)x(t) + D(\theta)u(t) + G(\theta), \quad (4.4)$$

where  $A, B, C, D, F, G$  matrices are with proper dimensions. These matrices are parameterized by a parameter vector  $\theta$ . The signals  $x(t), u(t), y(t)$  are not perturbation variables as are used in the conventional linearization models. The use of unperturbed signals facilitates model based analysis and the integration of multiple models into a global model.

Another motivation of considering CASS model as the local model structure for multimodels is the recent trends of controller design methods. The most natural place to find CASS model structure is in gain-scheduling control by off-equilibrium linearizations (Johansen et al., 1998). In contrast to the conventional gain-scheduling based on linearization about a set of equilibrium points, off-equilibrium scheduling is used to improve the transient performance as well as stability. Despite the improved results of off-equilibrium linearization, the design of the controller is still arbitrary and no analytical guarantee is presented.

In recent years, there have been increasing efforts to develop model based fuzzy control, in order to provide systematic controller design methods rather than heuristic ones using fuzzy inference. Since state space model has been the main stream in model based control, model based fuzzy control has also taken this approach. In (Kroll et al., 2000), fuzzy model based controller is proposed. The affine i/o TS fuzzy model is transformed into state space by defining the states as  $x(t) = [y(t-1), \dots, y(t-n_y), u(t-2), \dots, u(t-n_u)]^T$ . The state feedback controller is designed using pole-placement technique by considering the obtained state model as linear time-variant model. The steady state error caused from bias terms is compensated by adjusting the set-point by a set-point filter. However, placing all the poles of the closed-loop

system for time-varying systems does not necessarily guarantee the stability since stability is guaranteed only for slowly time varying systems (Shamma and Athans, 1990).

More theoretically sound approach is to utilize the extension of Lyapunov stability theorems to the multimodel framework (Tanaka and Sano, 1994). Basic approach is to find a positive definite matrix such that the quadratic Lyapunov candidate satisfies the stability conditions for the fuzzy model. That is, consider a fuzzy model without external input as

$$x(t+1) = \sum_{i=1}^m w_i(\varphi(t)) A_i x(t) \quad (4.5)$$

and it is shown that finding  $P > 0$  satisfying

$$A_i^T P A_i - P < 0 \quad (4.6)$$

is a sufficient condition for the stability of the origin (Tanaka and Sano, 1994). Just like Lyapunov theorem for nonlinear systems, this condition can be used for controller design with state feedback. In (Tanaka et al., 1996), similar approach is extended for fuzzy model with uncertainty model. Finding of positive definite matrices are solved by LMI (linear matrix inequality). However, all these works are based on linear fuzzy model such as (4.5) and only used for the controller design with available nonlinear models. The nonlinear model is converted into fuzzy models by adopting complex nonlinear fuzzy membership functions. Therefore, their works can only be used as an alternative to global nonlinear controller design. The problem of fuzzy model with linear local models is obvious for uncertain systems without an accurate nonlinear model available since identification of the complex nonlinear fuzzy membership functions that guarantee the validity of linear local models will be extremely difficult. Model based fuzzy controller design for uncertain systems reported in (Cao et al., 1995, 1997a) are also based on linear local models. Interestingly, Cao et al. (1997b), the first part of (Cao et al., 1997a) regarding system identification for controller de-

sign, used affine model, however, the affine term disappears in the controller design without any explanation. The first work considering affine models for controller design is reported in (Kiriakidis et al., 1998), which takes the similar approach, i.e. extended Lyapunov stability and LMI based design. However, the affine state space model is obtained in a heuristic way, as pointed out in Chapter 3.

From the overview of the current model based fuzzy control literatures, we can see the demand of identification of CASS model as a local model. Justification of using linear models in the literatures is to adopt complex nonlinear fuzzy membership functions which is hard to find in system identification.

### 4.3 CASS model structure for identification

From the previous section, the motivation of CASS model for control is clear. In this section, we investigate the CASS model structure from system identification perspectives.

Before the estimation of parameters, it is essential to examine if the free parameters of the model can be uniquely recovered from measurements unless our interest of the model is only in the duplication of i/o relationship of the system in the training data set. This uniqueness of parameterization is called *structural identifiability*. Great emphasis has been placed on the structural identifiability related to the gray box type identification in biological or chemical applications, whose model parameters have physical interpretations (Walter and Pronzato, 1990). The formal definition of identifiability is given as follows (Ljung, 1999).

**Definition 4.1 (Global structural identifiability)** A model structure  $M$  is globally identifiable at  $\theta^*$  if the equivalence of a model structure

$$M(\theta) = M(\theta^*),$$

implies

$$\theta = \theta^*.$$

**Definition 4.2 (Local structural identifiability)** The model structure is locally identifiable if for  $\theta^*$  there exists a neighborhood  $B(\theta^*, \epsilon)$  such that

$$M(\theta) = M(\theta^*), \theta \in B(\theta^*, \epsilon)$$

implies

$$\theta = \theta^*.$$

In (Ljung, 1999), the equivalence of a model structure implies the equivalence of transfer functions. This is only possible for linear systems, since the system is separable from signals in transformed space. For nonlinear systems, the equivalence implies for the whole output space as the definition given in Chapter 3. Among the same model structures, the definition of equivalence given in Chapter 3 can be relaxed since the concept of initial conditions is clear. We adopt the definition of equivalence from (Vajda et al., 1989) and modify it for a discrete time system.

**Definition 4.3 (Equivalence of systems in state space (Vajda et al., 1989))**

Consider a system given in (4.1)-(4.2) and denote the input-output map as

$$\Sigma_{\theta}^{x_0}(u) : u(\cdot) \rightarrow y(\cdot)$$

where  $\theta$  is the parameter vector and  $x_0$  is the initial condition. We define that the systems are equivalent if and only if

$$\Sigma_{\theta}^{x_0}(u) = \Sigma_{\bar{\theta}}^{x_0}(u) \tag{4.7}$$

Using the Definition 4.3, we can obtain the following result for CASS model.

**Proposition 4.1 (Necessary condition of equivalence of CASS model)** Consider the CASS models given as follows:

$$\begin{aligned} \Sigma : \quad x(t+1) &= Ax(t) + Bu(t) + F \\ y(t) &= Cx(t) + Du(t) + G \end{aligned} \quad (4.8)$$

$$\begin{aligned} \bar{\Sigma} : \quad \bar{x}(t+1) &= \bar{A}\bar{x}(t) + \bar{B}u(t) + \bar{F} \\ \bar{y}(t) &= \bar{C}\bar{x}(t) + \bar{D}u(t) + \bar{G} \end{aligned} \quad (4.9)$$

$\bar{x}, x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}$  and  $y \in Y \subset \mathbb{R}$ . The system matrices have proper dimensions accordingly. The necessary condition for equivalence is the existence of affine transformation between  $x(t)$  and  $\bar{x}(t)$  such that

$$x(t) = T\bar{x}(t) + w \quad (4.10)$$

where

$$T = \Gamma_r^\dagger \bar{\Gamma}_r \quad (4.11)$$

$$w = \Gamma_r^\dagger (\bar{\Lambda}_r \bar{F} + \mathbf{1}_r \bar{G} - \Lambda_r F - \mathbf{1}_r G). \quad (4.12)$$

where  $()^\dagger$  denotes pseudo-inverse. The matrices in  $w$  is defined as

$$\Gamma_r = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix} \quad (4.13)$$

$$\Lambda_r = \begin{bmatrix} 0 \\ C \\ CA + C \\ \vdots \\ CA^{r-2} + CA^{r-3} + \dots + C \end{bmatrix} \quad (4.14)$$

$$\mathbf{1}_r = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad (4.15)$$

with some positive integer  $r \geq n$ .

**Proof:**

Construct an augmented output vector of  $\Sigma$  as

$$\begin{aligned}
Y_r &= \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(r-1) \end{bmatrix} \\
&= \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix} x(0) \\
&+ \begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & & & \\ CA^{r-2}B & \dots & \dots & D \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(r-1) \end{bmatrix} \\
&+ \begin{bmatrix} 0 \\ C \\ CA+C \\ \vdots \\ CA^{r-2} + CA^{r-3} + \dots + C \end{bmatrix} F + \begin{bmatrix} G \\ G \\ \vdots \\ G \end{bmatrix}, \\
&= \Gamma_r x(0) + H_r U_r + \Lambda_r F + \mathbf{1}_r G
\end{aligned}$$

Similarly, the augmented output vector of  $\bar{\Sigma}$  is given as,

$$\bar{Y}_r = \bar{\Gamma}_r \bar{x}(0) + \bar{H}_r U_r + \bar{\Lambda}_r \bar{F} + \bar{\mathbf{1}}_r \bar{G} \quad (4.16)$$

The equivalence of the two systems implies  $Y_r = \bar{Y}_r$ . Since  $u(t)$  is arbitrary,  $H_r = \bar{H}_r$  and therefore,

$$x(0) = \Gamma_r^\dagger \bar{\Gamma}_r x(0) + \Gamma_r^\dagger (\bar{\Lambda}_r \bar{F} + \bar{\mathbf{1}}_r \bar{G} - \Lambda_r F - \mathbf{1}_r G). \quad (4.17)$$

Since the coordinate of  $x(t)$  is fixed by this initial state transformation,

$$x(t) = T \bar{x}(t) + w \quad (4.18)$$

where

$$T = \Gamma_r^\dagger \bar{\Gamma}_r \quad (4.19)$$

$$w = \Gamma_r^\dagger (\bar{\Lambda}_r \bar{F} + \bar{\mathbf{1}}_r \bar{G} - \Lambda_r F - \mathbf{1}_r G). \quad (4.20)$$



□

Since state space representation is not unique, we consider the CASS model in the canonical observable form in order to reduce the dimension of parameter vector for the facilitation of analysis. Conjectured from linear systems, we propose the similarity transformation matrix to obtain the observable canonical form such as  $x(t) = T\bar{x}(t)$ . The following proposition shows that the i/o relations are invariant to linear similarity transformation.

**Proposition 4.2 (Sufficient condition of equivalence of CASS model)** Consider the CASS model represented as (4.3)-(4.4). Then the i/o relationships are invariant to the similarity transformation  $x(t) = T\bar{x}(t)$  with invertible matrix  $T$ . That is, the existence of linear similarity transformation is the sufficient condition of equivalence of two CASS models.

**Proof:** Replace  $x(t)$  with  $T\bar{x}(t)$  where  $\bar{x}(t)$  is the newly defined coordinates by similarity transformation matrix. Then, the CASS model becomes

$$\bar{x}(t+1) = T^{-1}AT\bar{x}(t) + T^{-1}Bu(t) + T^{-1}F \quad (4.21)$$

$$y(t) = CT\bar{x}(t) + Du(t) + G \quad (4.22)$$

with new initial condition of  $\bar{x}(0) = T^{-1}x(0)$ . Define the new CASS model matrices as  $\bar{A} = T^{-1}AT$ ,  $\bar{B} = T^{-1}B$ ,  $\bar{F} = T^{-1}F$ ,  $\bar{C} = CT$ ,  $\bar{D} = D$ , and  $\bar{G} = G$ . Then the i/o relationship of transformed states can be represented as

$$y(0) = \bar{C}\bar{x}(0) + Du(0) + G \quad (4.23)$$

$$= CTT^{-1}x(0) + Du(0) + G \quad (4.24)$$

$$= Cx(0) + Du(0) + G \quad (4.25)$$

at  $t = 0$ . Hence,  $(u(0), y(0))$  is not changed. For an arbitrary  $t$ ,

$$y(t) = \bar{C}\bar{A}^{t-1}\bar{x}(0) + \sum_{i=1}^{t-1} \bar{C}\bar{A}^{i-1}\bar{B}u(t-i) + Du(t) + \sum_{i=1}^{t-1} \bar{C}\bar{A}^{i-1}\bar{F} + \bar{G}. \quad (4.26)$$

It is obvious that the similarity transformations of linear portion cancel out. For the affine terms,

$$\sum_{i=1}^{t-1} \bar{C}\bar{A}^{i-1}\bar{F} + \bar{G} = \sum_{i=1}^{t-1} CTT^{-1}A^{i-1}TT^{-1}F + G \quad (4.27)$$

Hence, we can see that the affine terms also do not change through similarity transformation matrix.  $\square$

The realization of canonical form is always possible if the system states are minimal. Also, Proposition 4.2 guarantees the existence of transformation to the observable canonical form. The definition of minimal state realization of nonlinear systems varies depending on the use of the model. In this dissertation, we adopt a strong minimality condition as follows.

**Definition 4.4 (Minimal state space)** The nonlinear system given in (4.1)-(4.2) is minimal if  $(A, C)$  is observable and  $(A, B)$  is controllable.

Hence, the minimality of nonlinear state space equation is completely determined by the minimality of linear portion of local model, i.e. CASS model. The CASS

model in the observable canonical form has the parameterization as

$$A(\theta) = \begin{bmatrix} a_1 & 1 & & & \\ a_2 & 0 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ a_{n-1} & 0 & 0 & \dots & 1 \\ a_n & 0 & 0 & \dots & 0 \end{bmatrix} \quad (4.28)$$

$$B(\theta) = [ b_1 \ b_2 \ \dots \ b_n ]^T \quad (4.29)$$

$$C(\theta) = [ 1 \ 0 \ \dots \ 0 ] \quad (4.30)$$

$$D(\theta) = [ d ] \quad (4.31)$$

$$F(\theta) = [ f_1 \ f_2 \ \dots \ f_n ]^T \quad (4.32)$$

$$G(\theta) = [ g ] \quad (4.33)$$

In general, it is difficult to verify the identifiability of a model structure by the definition. For linear systems, similarity transformation approach is given in (Walter and Pronzato, 1990) for state space models. For nonlinear systems, structural identifiability is based on the unique solvability of the parameters. General i/o equation is reduced by using Ritt's algorithm in order to single out parameters in terms of i/o and their derivatives (Ljung and Glad, 1994). Among various methods, Taylor series approach, utilizing the uniqueness of the coefficients of Taylor series, and extended similarity transformation, which is an extension of the similarity transformation results in linear systems by local/global isomorphism, are promising methods (Chappell et al., 1999; Chappell and Godfrey, 1990). Now, the dilemma that we face is that CASS model structure is too simple to be considered as nonlinear model structure while we cannot apply linear techniques since it is not linear model structure. We propose a method to check the identifiability of CASS model. The method is an extension of similarity transformation method in linear systems rather than the ones in nonlinear systems. First, we adopt a more clear definition of identifiability and modify it for a discrete time system, since the formal definitions given by (Ljung, 1999) is too abstract. Structural identifiability is similar to equivalence, except that the

uniqueness of parameters has to be satisfied as well as the uniqueness of i/o relation.

**Definition 4.5 (Modified structural identifiability (Vajda et al., 1989))** A model is *globally identifiable* at  $\theta \in \Theta$  if and only if

$$\Sigma_{\theta}^{x_0(\theta)}(u) = \Sigma_{\bar{\theta}}^{x_0(\bar{\theta})}(u) \quad (4.34)$$

for all  $u \in [0, t_1]$  implies  $\theta = \bar{\theta}$ . It is *locally identifiable* at  $\theta \in \Theta$  if there is some neighborhood  $\bar{\theta} \in \bar{\Theta} \subset \Theta$  such that (4.34) implies  $\theta = \bar{\theta}$ .

**Theorem 4.1 (Identifiability condition of CASS model)** CASS model in observable form is identifiable if

$$\begin{bmatrix} I_n - (A - I_n)\Gamma^\dagger\Lambda & (I_n - A)\Gamma^\dagger\mathbf{1} \\ -C\Gamma^\dagger\Lambda & 1 - C\Gamma^\dagger\mathbf{1} \end{bmatrix} \quad (4.35)$$

is not singular.

**Proof:** Suppose two CASS models in observable canonical form are equivalent. Then, from Proposition 4.1, it implies that there is an affine transformation as

$$x(t) = T\bar{x}(t) + w. \quad (4.36)$$

Therefore, we have to show that  $\Sigma_{\theta}^{x_0(\theta)} = \bar{\Sigma}_{\theta^*}^{\bar{x}_0(\theta^*)}$ , where  $\Sigma$  and  $\bar{\Sigma}$  are related by the affine transformation, implies  $\theta = \theta^*$ .

Replace  $x(t)$  with (4.36) and we obtain

$$T\bar{x}(t+1) + w = A(T\bar{x}(t) + w) + Bu(t) + F \quad (4.37)$$

which can be simplified as

$$\bar{x}(t+1) = T^{-1}AT\bar{x}(t) + T^{-1}Bu(t) + T^{-1}(F + (A - I)w) \quad (4.38)$$

$$= \bar{A}\bar{x}(t) + \bar{B}u(t) + \bar{F}. \quad (4.39)$$

The transformed output is

$$y(t) = C(T\bar{x}(t) + w) + Du(t) + G \quad (4.40)$$

$$= CT\bar{x}(t) + Du(t) + G + Cw \quad (4.41)$$

$$= \bar{C}\bar{x}(t) + Du(t) + \bar{G}. \quad (4.42)$$

Hence, it is obvious that the following holds

$$\bar{F} = T^{-1}(F + (A - I)w) \quad (4.43)$$

$$\bar{G} = G + Cw \quad (4.44)$$

in addition to the usual linear portion with similarity transformation  $\bar{A} = T^{-1}AT$ ,  $\bar{B} = T^{-1}B$ ,  $\bar{C} = CT$ , and  $\bar{D} = D$ .

Consider a 2nd order CASS system as

$$x(t+1) = A(\theta)x(t) + B(\theta)u(t) + F(\theta)$$

$$y(t) = C(\theta)x(t) + D(\theta)u(t) + G(\theta)$$

where  $A(\theta) = \begin{bmatrix} a_1 & 1 \\ a_2 & 0 \end{bmatrix}$ ,  $B(\theta) = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ ,  $F(\theta) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ ,  $C(\theta) = [1 \ 0]$ ,  $D(\theta) = d$ , and  $G(\theta) = g$ . Suppose that there is another parameter vector  $\theta^*$ , such that  $A(\theta^*) = T^{-1}A(\theta)T$ ,  $B(\theta^*) = T^{-1}B(\theta)$ ,  $F(\theta^*) = T^{-1}F(\theta)$ ,  $C(\theta^*) = C(\theta)T$ , and  $D(\theta^*) = D(\theta)$  for a similarity transformation matrix  $T$ .

Since  $C(\theta^*) = C(\theta)T$ ,

$$[1 \ 0] = [1 \ 0] \begin{bmatrix} t_{1,1} & t_{1,2} \\ t_{2,1} & t_{2,2} \end{bmatrix}, \quad (4.45)$$

which results in  $t_{1,1} = 1$  and  $t_{1,2} = 0$ . With  $TA(\theta^*) = A(\theta)T$ ,

$$\begin{bmatrix} 1 & 0 \\ t_{2,1} & t_{2,2} \end{bmatrix} \begin{bmatrix} a_1^* & 1 \\ a_2^* & 0 \end{bmatrix} = \begin{bmatrix} a_1 & 1 \\ a_2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_{2,1} & t_{2,2} \end{bmatrix}, \quad (4.46)$$

the resulting matrix is

$$\begin{bmatrix} a_1^* & 1 \\ a_1^*t_{2,1} + a_2^*t_{2,2} & t_{2,1} \end{bmatrix} = \begin{bmatrix} a_1 + t_{2,1} & t_{2,2} \\ a_2 & 0 \end{bmatrix}.$$

Hence,  $t_{2,1} = 0$  and  $t_{2,2} = 1$ , and therefore  $T = I_2$ , where  $I_2$  is the  $2 \times 2$  identity matrix, and it implies  $\theta = \theta^*$ . Therefore, we can see that the similarity transformation is uniquely determined by  $A$  and  $C$  matrices, and  $F, G$  matrices do not influence  $T$ . Extend this result to the general  $n$ th order system. By the generalization of (4.45), it is clear that  $t_{1,1} = 1$ ,  $t_{1,i} = 0, i = 2, \dots, n$ . From the generalization of (4.46),

$$\begin{bmatrix} a_1^* & 1 & 0 & \dots & 0 \\ \sum t_{2,i}a_i^* & t_{2,1} & t_{2,2} & \dots & t_{2,n-1} \\ \sum t_{3,i}a_i^* & t_{3,1} & t_{3,2} & \dots & t_{3,n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum t_{n-1,i}a_i^* & t_{n-1,1} & t_{n-1,2} & \dots & t_{n-1,n-1} \\ \sum t_{n,i}a_i^* & t_{n,1} & t_{n,2} & \dots & t_{n,n-1} \end{bmatrix} = \begin{bmatrix} a_1 + t_{2,1} & t_{2,2} & t_{2,3} & \dots & t_{2,n} \\ a_2 + t_{3,1} & t_{3,2} & t_{3,3} & \dots & t_{3,n} \\ a_3 + t_{4,1} & t_{4,2} & t_{4,3} & \dots & t_{4,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-1} + t_{n,1} & t_{n,2} & t_{n,3} & \dots & t_{n,n} \\ a_n & 0 & 0 & \dots & 0 \end{bmatrix},$$

where  $i = 1, \dots, n$ . Solve the above equation for the elements of  $T$ , and we obtain  $T = I_n$ . Therefore,  $\theta = \theta^*$  for linear matrices  $A, B, C, D$ . Since  $T = I_n$ , the bias conditions can also be simplified as

$$\bar{F} = F + (A - I)\Gamma^\dagger(\Lambda(\bar{F} - F) + \mathbf{1}(\bar{G} - G)) \quad (4.47)$$

$$\bar{G} = G + C\Gamma^\dagger(\Lambda(\bar{F} - F) + \mathbf{1}(\bar{G} - G)). \quad (4.48)$$

Simplify the above equations, then

$$\begin{bmatrix} I_n - (A - I)\Gamma^\dagger\Lambda & (I_n - A)\Gamma^\dagger\mathbf{1} \\ -C\Gamma^\dagger\Lambda & \mathbf{1} - C\Gamma^\dagger\mathbf{1} \end{bmatrix} \begin{bmatrix} \bar{F} - F \\ \bar{G} - G \end{bmatrix} = 0 \quad (4.49)$$

Hence if the matrix in the l.h.s. is not singular, then

$$\bar{F} - F = 0, \bar{G} - G = 0 \quad (4.50)$$

which implies structural identifiability of CASS model in observable canonical form.

□

Even though the theorem results in a simple condition to check the structural identifiability, pseudo-inverse of  $\Gamma$  necessitates the use of symbolic computational software. Simpler method can be derived by extending the transfer function method for linear systems to affine systems.

**Proposition 4.3 (Structural Identifiability of CASS model)** CASS model is not identifiable from input-output.

**Proof:** From the sufficient condition of the equivalence of CASS model, we can transform any CASS model into observable form. Then, by taking the  $z$ -transformation of the CASS model with parameterization (4.28) through (4.33) in addition to the parameterized initial states. The transfer function becomes

$$\begin{aligned}
Y(z) &= \left[ \frac{b_1 z^{n-1} + b_2 z^{n-2} + \dots + b_n}{z^n - a_1 z^{n-1} - a_2 z^{n-2} - \dots - a_n} + d \right] U(z) \\
&+ \left[ \frac{f_1 z^{n-1} + f_2 z^{n-2} + \dots + f_n}{z^n - a_1 z^{n-1} - a_2 z^{n-2} - \dots - a_n} + g \right] \frac{z}{z-1} \\
&+ \left[ \frac{x_{01} z^{n-1} + \dots + x_{0n}}{z^n - a_1 z^{n-1} - a_2 z^{n-2} - \dots - a_n} \right] z.
\end{aligned}$$

Combine and simplify the equation

$$\begin{aligned}
Y(z) &= \left[ \frac{dz^n + (b_1 - a_1 d)z^{n-1} + \dots + (b_n - a_n d)}{z^n - a_1 z^{n-1} - a_2 z^{n-2} - \dots - a_n} \right] U(z) \\
&+ \left[ \frac{(g + x_{01})z^n + (f_1 - a_1 g + x_{02} - x_{01})z^{n-1}}{z^n - a_1 z^{n-1} - a_2 z^{n-2}} \right. \\
&\quad \left. + \dots + \frac{(f_n - a_n g - x_{0n})}{-\dots - a_n} \right] \frac{z}{z-1} \tag{4.51}
\end{aligned}$$

$$= G(z; \theta)U(z) + H(z; \theta) \frac{z}{z-1} \tag{4.52}$$

Similarly, the i/o relation of parameter vector  $\bar{\theta}$  can also be represented as

$$\bar{Y}(z) = G(z; \bar{\theta})U(z) + H(z; \bar{\theta}) \frac{z}{z-1}. \tag{4.53}$$

Since  $U(z)$  is arbitrary,  $G(z; \theta) = G(z; \bar{\theta})$  and this implies  $H(z; \theta) = H(z; \bar{\theta})$ . Since the denominator of  $H(z; \theta)$  is same as  $H(z; \bar{\theta})$  and the numerators of  $H(z; \theta)$  and  $H(z; \bar{\theta})$  are polynomial functions, the numerators must be same. Since the numerators are  $n$ th order polynomials with  $2n + 1$  coefficients, observable CASS model is not structurally identifiable.  $\square$

**Remark 4.1** *In some literatures, initial conditions are not explicitly considered in the definition of structural identifiability (Van Den Hof, 1998). This may be contributed by the fact that the effect of initial states decays away for stable systems, and structural identifiability has been usually considered for gray box identification, which implies a fair amount of a priori knowledge about the system. In the above proposition, CASS model is structurally identifiable without initial states. However, assuming known initial states in multimodels can be a very restrictive assumption.*

## 4.4 Significance of unidentifiability of CASS model

The unidentifiability property of CASS model shown in the previous section is quite a discouraging result. Without identifiability, the model cannot be recovered uniquely, which results in the failure of identification. Hence, with the problem of recursive i/o model noticed in Chapter 3 and problem of unidentifiability of state space fuzzy model, the chance of justifying multimodel approaches to identification and control by rigorous way is slim. In this section, we investigate the significance of the unidentifiability from different perspectives by considering several popular identification methods.

### 4.4.1 Optimization method

In linear as well as nonlinear system identification, optimization based approach is the most popular one because of its general applicabilities regardless of model structures, and good convergence of model i/o to the measured i/o data. In this subsection, we consider the identification of CASS model by solving nonlinear least squares. We derive sensitivity functions of the model such that nonlinear least squares can be used. We illustrate the problem of blindly applying optimization method for the identification without verifying the identifiability of the model.

Optimization problem is to find parameters that minimize a cost function. The



application of optimization method to system identification is by minimizing the differences between the system and the model. Since the system is unknown, the difference is usually described by the outputs of the system and the model. One of the most common choices of the error measure is the summed squared error such as

$$\begin{aligned} V(\theta) &= \frac{1}{2N} \sum_{t=0}^{N-1} \epsilon(t|\theta)^2 \\ &= \frac{1}{2N} \sum_{t=0}^{N-1} (y(t) - \hat{y}(t|\theta))^2, \end{aligned} \quad (4.54)$$

or in a vector form

$$V(\theta) = \frac{1}{2N} E(\theta)^T E(\theta), \quad (4.55)$$

where  $y(t)$  is the measured output,  $\hat{y}(t|\theta)$  is the model output,  $\epsilon(t|\theta) = y(t) - \hat{y}(t|\theta)$ , and  $E(\theta) = [\epsilon(0|\theta), \epsilon(1|\theta), \dots, \epsilon(N-1|\theta)]^T$ . The minimization using this measure is called the Least Squares (LS) problem. We summarized the basic concept of nonlinear least squares in the Appendix A for the completeness of the presentation.

To update parameters using nonlinear least squares, the computation of gradient and/or Hessian of (4.54) is essential. For dynamical systems, the calculation of these derivatives can be complicated since the effect of parameter changes propagates through states. One way to see this propagation effect is by considering the following equation with parameter  $\theta$ ,

$$z(k+1) = f(\theta, z(k)), k = 1, 2, 3.$$

In the above equation,  $z(3)$  can be rewritten as

$$z(3) = z_3(\theta, z_2(\theta, z_1(\theta))),$$

where  $z_k$  represents the state at  $k$ . In order to compute the first derivative of  $z(3)$  with respect to  $\theta$ , we have to apply the chain rule up to  $z_1$ . The representation of this process can be simplified by using the so-called ordered partial derivative defined as

in (Piché, 1994),

$$\frac{\partial z_j}{\partial z_i} = \left. \frac{\partial^+ z_j}{\partial z_i} \right|_{(z_0, \dots, z_{i-1}) \text{ held constant}}, \quad (4.56)$$

where  $z_0$  denotes the parameter  $\theta$ . Then for the above exemplary equation,

$$\frac{\partial^+ z_3}{\partial z_0} = \frac{\partial z_3}{\partial z_0} + \frac{\partial z_3}{\partial z_2} \frac{\partial^+ z_2}{\partial z_0}, \quad (4.57)$$

where

$$\frac{\partial^+ z_2}{\partial z_0} = \frac{\partial z_2}{\partial z_0} + \frac{\partial z_2}{\partial z_1} \frac{\partial^+ z_1}{\partial z_0}, \quad (4.58)$$

and

$$\frac{\partial^+ z_1}{\partial z_0} = \frac{\partial z_1}{\partial z_0}. \quad (4.59)$$

Another way to implement the chain rule and the ordered derivatives is as

$$\frac{\partial^+ z_3}{\partial z_0} = \frac{\partial z_3}{\partial z_0} + \frac{\partial^+ z_3}{\partial z_1} \frac{\partial z_1}{\partial z_0} + \frac{\partial^+ z_3}{\partial z_2} \frac{\partial z_2}{\partial z_0}, \quad (4.60)$$

where

$$\frac{\partial^+ z_3}{\partial z_1} = \frac{\partial^+ z_3}{\partial z_2} \frac{\partial z_2}{\partial z_1}, \quad (4.61)$$

and

$$\frac{\partial^+ z_3}{\partial z_2} = \frac{\partial z_3}{\partial z_2}. \quad (4.62)$$

Since the sensitivity functions propagate forward in (4.57),(4.58),(4.59), this method is called the forward method. The propagation by (4.60),(4.61) and (4.62) is called the backward method for the same reason. Now, let us represent the partials of (4.54) with the ordered partial derivative for the model (4.3) and (4.4) parameterized by (4.28) through (4.33). First, consider the forward method.

The first partial derivatives of (4.54) are as

$$\frac{\partial V(\theta)}{\partial \theta_i} = -\frac{1}{N} \sum_{t=0}^{N-1} (y(t) - \hat{y}(t, \theta)) \frac{\partial^+ \hat{y}(t, \theta)}{\partial \theta_i}, \quad (4.63)$$

$$\frac{\partial^+ \hat{y}(t, \theta)}{\partial \theta_i} = \frac{\partial \hat{y}(t, \theta)}{\partial \theta_i} + \frac{\partial \hat{y}(t, \theta)^T}{\partial \hat{x}(t)} \frac{\partial^+ \hat{x}(t, \theta)}{\partial \theta_i}, \quad (4.64)$$

$$\frac{\partial^+ \hat{x}(t, \theta)}{\partial \theta_i} = \frac{\partial \hat{x}(t, \theta)}{\partial \theta_i} + \frac{\partial \hat{x}(t, \theta)}{\partial \hat{x}(t-1, \theta)} \frac{\partial^+ \hat{x}(t-1, \theta)}{\partial \theta_i}. \quad (4.65)$$

It is clear that  $\frac{\partial \hat{x}(t, \theta)}{\partial \hat{x}(t-1)} = A$  and  $\frac{\partial \hat{y}(t, \theta)}{\partial \hat{x}(t)} = C^T$ . The ordered partials of  $\hat{x}(t, \theta)$  for the parameterization by (4.28) through (4.33) is given as

$$\frac{\partial^+ \hat{x}(t+1)}{\partial a} = I_n \hat{x}_1(t) + A \frac{\partial^+ \hat{x}(t)}{\partial a} \quad (4.66)$$

$$\frac{\partial^+ \hat{x}(t+1)}{\partial B} = I_n u(t) + A \frac{\partial^+ \hat{x}(t)}{\partial B} \quad (4.67)$$

$$\frac{\partial^+ \hat{x}(t+1)}{\partial F} = I_n + A \frac{\partial^+ \hat{x}(t)}{\partial F} \quad (4.68)$$

$$\frac{\partial^+ \hat{x}(t+1)}{\partial x(0)} = A \frac{\partial^+ \hat{x}(t)}{\partial x(0)}, \quad (4.69)$$

where  $I_n$  is the  $n \times n$  identity matrix,  $a = [a_1, \dots, a_n]$  and  $x(0) = [x_1(0), \dots, x_n(0)]$ .

$\frac{\partial^+ \hat{x}(t)}{\partial \theta} = [\frac{\partial^+ \hat{x}(t)}{\partial \theta_1}, \frac{\partial^+ \hat{x}(t)}{\partial \theta_2}, \dots, \frac{\partial^+ \hat{x}(t)}{\partial \theta_n}]$ . The partials of  $\hat{y}(t, \theta)$  is represented as

$$\frac{\partial \hat{y}(t)}{\partial a} = \left[ \frac{\partial^+ \hat{x}(t)}{\partial a} \right]^T C^T \quad (4.70)$$

$$\frac{\partial \hat{y}(t)}{\partial B} = \left[ \frac{\partial^+ \hat{x}(t)}{\partial B} \right]^T C^T \quad (4.71)$$

$$\frac{\partial \hat{y}(t)}{\partial F} = \left[ \frac{\partial^+ \hat{x}(t)}{\partial F} \right]^T C^T \quad (4.72)$$

$$\frac{\partial \hat{y}(t)}{\partial x(0)} = \left[ \frac{\partial^+ \hat{x}(t)}{\partial x(0)} \right]^T C^T \quad (4.73)$$

$$\frac{\partial \hat{y}(t)}{\partial D} = u(t) \quad (4.74)$$

$$\frac{\partial \hat{y}(t)}{\partial G} = 1. \quad (4.75)$$

Here,  $\frac{\partial \hat{y}(t)}{\partial \theta}$  is the gradient vector.

The backward method can be derived by utilizing the ordered derivatives of  $V(\theta)$  with respect to  $\hat{x}(t)$ . That is,

$$\frac{\partial V(\theta)}{\partial \theta_i} = \left[ \frac{\partial^+ V(\theta)}{\partial \hat{x}(t)} \right]^T \frac{\partial \hat{x}(t)}{\partial \theta_i}.$$

For the least square cost function (4.54),

$$\frac{\partial^+ V}{\partial \hat{x}(t)} = \frac{\partial V}{\partial \hat{x}(t)} + \frac{\partial \hat{x}(t+1)}{\partial \hat{x}(t)}^T \frac{\partial^+ V}{\partial \hat{x}(t+1)} \quad (4.76)$$

$$= -(y(t) - \hat{y}(t)) C^T + A^T \frac{\partial^+ V}{\partial \hat{x}(t+1)} \quad (4.77)$$

for  $t = 0, \dots, N - 1$  with initial condition

$$\frac{\partial^+ V}{\partial \hat{x}(N)} = 0. \quad (4.78)$$

For the CASS model parameters, (4.28) through (4.33),

$$\frac{\partial V}{\partial a} = \frac{1}{N} \sum_{t=0}^{N-1} \frac{\partial \hat{x}(t)^T}{\partial a} \frac{\partial^+ V}{\partial \hat{x}(t)} \quad (4.79)$$

$$\frac{\partial V}{\partial B} = \frac{1}{N} \sum_{t=0}^{N-1} \frac{\partial \hat{x}(t)^T}{\partial B} \frac{\partial^+ V}{\partial \hat{x}(t)} \quad (4.80)$$

$$\frac{\partial V}{\partial F} = \frac{1}{N} \sum_{t=0}^{N-1} \frac{\partial \hat{x}(t)^T}{\partial F} \frac{\partial^+ V}{\partial \hat{x}(t)} \quad (4.81)$$

and

$$\frac{\partial V}{\partial D} = -\frac{1}{N} \sum_{t=0}^{N-1} (y(t) - \hat{y}(t))u(t) \quad (4.82)$$

$$\frac{\partial V}{\partial G} = -\frac{1}{N} \sum_{t=0}^{N-1} (y(t) - \hat{y}(t)) \quad (4.83)$$

where

$$\frac{\partial \hat{x}(t)}{\partial a} = I_n \hat{x}_1(t-1) \quad (4.84)$$

$$\frac{\partial \hat{x}(t)}{\partial B} = I_n u(t-1) \quad (4.85)$$

$$\frac{\partial \hat{x}(t)}{\partial F} = I_n. \quad (4.86)$$

By comparing the forward method with the backward method, we have to solve recurrent equations for each parameter related to the state equation as shown in (4.66) through (4.69), which are  $4n \times n$  recurrent equations. In contrast, the backward method only requires the recurrent equation (4.77), which are  $n$  recurrent equations. Hence, when only the efficiency of the gradient computation is considered, the backward method is advantageous. The problem is the forward method leads to the Gauss-Newton method or the Levenberg-Marquardt method, which is much more efficient than steepest-descent method in the convergence rate. Hence, we can

conclude that the forward method will lead to more efficient overall convergence than the backward method.

It is straightforward to apply nonlinear least squares, once sensitivity functions are available. However, unique estimation of unknown parameters is not valid because of the unidentifiability of CASS model structure. We illustrate this with the following example.

**Example 4.1** *Consider the following CASS model as the data generating system from Chapter 3,*

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) + F \\y(t) &= Cx(t) + Du(t) + G,\end{aligned}$$

where  $x(t) = [x_1(t) \ x_2(t)]^T$ ,  $A = \begin{bmatrix} 0.965 & 0.0975 \\ -0.689 & 0.916 \end{bmatrix}$ ,  $B = \begin{bmatrix} 0.005 \\ 0.0975 \end{bmatrix}$ ,  $C = [ -0.456 \ 0 ]$ ,  $D = 0$ ,  $F = \begin{bmatrix} -0.00791 \\ -0.1483 \end{bmatrix}$  and  $G = 0.814$ .

*A data set with a length of 1000 is obtained by exciting the system with Gaussian random input with mean 7 and unit variance. The nonlinear LS based algorithm is implemented using the Levenberg-Marquardt algorithm with the simple Levenberg-Marquardt parameter update rule by the multiplication/division factor. The trust-region type method was also implemented (Kelley, 1999), however, it did not show much difference in the convergence rate. Initial parameters are chosen randomly.*

*Instead of converging into the true parameters, the nonlinear least squares based algorithm converges to the parameter sets to satisfy (4.51) in Proposition 4.3. Hence, we can interpret the property of structural identifiability into the existence of global optimum. We believe that the observation of this result is important since bias terms are commonly used in nonlinear black box model structures.*

## 4.4.2 Input-output based method

From this subsection to the rest of this chapter, we only focus on the identification of the bias terms. This is motivated from the observation of the behavior of the optimization based method. In the optimization based method, the model parameters of the linear portion are uniquely recovered from data while the bias terms are incorrectly estimated. The estimation of the linear portion can be easily done by introducing new variables as  $\delta x(t) = x(t+1) - x(t)$ ,  $\delta u(t) = u(t+1) - u(t)$ ,  $\delta y(t) = y(t+1) - y(t)$ , and take the differences of (4.3) and (4.4) respectively, then we obtain

$$\begin{aligned}\delta x(t+1) &= A\delta x(t) + B\delta u(t), \\ \delta y(t) &= C\delta x(t) + D\delta u(t).\end{aligned}$$

Now, we can apply subspace method to estimate  $A, B, C, D$  matrices. Since we consider a canonical form, the estimation can also be done by linear least squares.

From (4.3)-(4.4), we can obtain i/o equation as

$$y(t) - \left( \sum_{i=1}^{t-1} CA^{i-1}Bu(t-i) + Du(t) \right) = \left[ \sum_{i=1}^{t-1} CA^{i-1} \quad 1 \right] \begin{bmatrix} F \\ G \end{bmatrix} + CA^{t-1}x(0) \quad (4.87)$$

Suppose that the parameters of linear portion of CASS model are estimated from the procedure described above, the l.h.s of the equation is all known and this is a typical linear least square problem. Even though the estimation of linear matrices can contain estimation errors, we ignore the effects. Denote (4.87) as

$$z(t) = \varphi(t)^T \theta \quad (4.88)$$

where  $z(t) = y(t) - (\sum_{i=1}^{t-1} CA^{i-1}Bu(t-i) + Du(t))$ ,  $\varphi(t)^T = [ CA^{t-1} \quad \sum_{i=1}^{t-1} CA^{i-1} \quad 1 ]$  and  $\theta = [x(0)^T, F^T, G^T]^T$ . Then the matrix-vector representation of (4.87) is given as

$$Z = \Phi^T \theta \quad (4.89)$$

where  $Z$  and  $\Phi$  are augmented vector and matrix of  $z(t)$  and  $\varphi^T(t)$  at  $t = 0, \dots, N - 1$  with  $N$  the number of data. In order for (4.89) to have a unique solution,  $\Phi$  must have full column rank. We verify the condition of full column rank in the following.

**Proposition 4.4 (Rank deficiency of CASS model)**  $\Phi$  is rank deficient

**Proof:** The  $\Phi$  matrix is represented as

$$\Phi = \begin{bmatrix} C & 0 & 1 \\ CA & C & 1 \\ CA^2 & C + CA & 1 \\ \vdots & \vdots & \vdots \\ CA^{N-1} & \sum_{i=1}^{N-1} CA^{i-1} & 1 \end{bmatrix}. \quad (4.90)$$

From the second row, do the row manipulation by subtracting the previous row from the current row and we obtain

$$\Phi = \begin{bmatrix} C & 0 & 1 \\ C(A - I) & C & 0 \\ CA(A - I) & CA & 0 \\ \vdots & \vdots & \vdots \\ CA^{N-2}(A - I) & CA^{N-2} & 0 \end{bmatrix}. \quad (4.91)$$

From Cayley-Hamilton theorem,  $A^n = a_0I + \dots + a_nA^{n-1}$ . Hence, we can easily see that the rank of  $\Phi$  is  $n + 1$ .  $\square$

**Remark 4.2** *From the above proposition, we can see that the estimation of  $[F, G]$  is not unique because of rank deficiency.*

### 4.4.3 Bias estimation by augmented states

It is a common practice to augment the unknown biases as additional states and apply the well-known Kalman filters for the estimation of the states of a linear system with biases (Friedland, 1969). CASS model is indeed a linear state model with states/output biases and it might be natural to consider the augmented Kalman filter

as a viable solution for the bias estimation. The augmented system can be represented as

$$\begin{bmatrix} x(t+1) \\ z_1(t+1) \\ z_2(t+1) \end{bmatrix} = \begin{bmatrix} A & I_n & 0 \\ 0 & I_n & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ z_1(t) \\ z_2(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u(t) \quad (4.92)$$

$$y(t) = [C \ 0 \ I] \begin{bmatrix} x(t) \\ z_1(t) \\ z_2(t) \end{bmatrix} + Du(t). \quad (4.93)$$

The basic property that the augmented system has to satisfy is the observability. The result is as follows.

**Proposition 4.5 (Observability of augmented linear model from CASS model)**

The augmented linear system is not observable.

**Proof:** The observability matrix is same as  $\Phi$  matrix in Proposition 4.4 and the rank of the matrix is shown to be  $n + 1$ . □

**Remark 4.3** *In (Bembenek et al., 1998), the observability of augmented linear system for Kalman filters with state bias is shown. They also concluded that the augmented system is unobservable with single output but this can be relaxed with multiple outputs, which can be interpreted as a method for experiment design.*

#### 4.4.4 Bias estimation by robust observer

By now, the significance of unidentifiability of CASS model in system identification must be clear. In this subsection, we adopt quite a different approach. Motivated from control theories, we attempt to utilize observers to estimate states and use this state estimates for the estimation of states/output biases. The estimated states from observers are biased for affine systems since the states/output offset terms work as step disturbances. To reduce the bias caused by the affine terms, we investigate state estimation by robust observers. Since we are interested in off-line approach, we also



investigate if iteration of state estimation followed by offset estimation can improve the estimation of the offset terms.

Firstly, consider the Luenberger observer. Since Luenberger observer does not deal with disturbances explicitly, we can expect that the unknown states/output offsets will have significant effects on the state estimates. For simplicity, let  $D = 0$  and  $G = 0$ . That is,

$$x(t+1) = Ax(t) + Bu(t) + F \quad (4.94)$$

$$y(t) = Cx(t). \quad (4.95)$$

Then, Luenberger observer is represented as

$$\hat{x}^i(t+1) = A\hat{x}^i(t) + Bu(t) + F + L(y(t) - \hat{y}^i(t)) \quad (4.96)$$

$$\hat{y}^i(t) = C\hat{x}^i(t). \quad (4.97)$$

where  $L$  is the observer gain. The superscript  $i$  in  $\hat{y}^i(t)$  and  $\hat{x}^i(t)$  is index for iteration, which will become clear shortly. Ignore the estimation error of  $A$ ,  $B$  and  $C$  matrices and only consider that  $F$  is uncertain. Since  $F$  is not certain, replace  $F$  in the (4.96) with  $\hat{F}^i$ . Then, by subtracting (4.96) from (4.94) and by defining estimated states error  $\tilde{x}^i(t) = x(t) - \hat{x}^i(t)$ , we obtain an observer error dynamics as

$$\begin{aligned} \tilde{x}^i(t+1) &= A\tilde{x}^i(t) + \tilde{F}^i - L(y(t) - \hat{y}^i(t)) \\ &= (A - LC)\tilde{x}^i(t) + \tilde{F}^i. \end{aligned} \quad (4.98)$$

If the observer gain  $L$  is chosen such that  $A - LC$  is stable, in steady state, the states estimate error is  $\tilde{x}^i(t) = \tilde{x}^i(\infty)$  and

$$\begin{aligned} (I - A + LC)\tilde{x}^i(\infty) &= \tilde{F}^i \\ \tilde{x}^i(\infty) &= (I - A + LC)^{-1}\tilde{F}^i. \end{aligned} \quad (4.99)$$

Now, we may wonder how to improve the estimate of  $F$  from  $\hat{x}^i(t)$ . From (4.94), a reasonable choice would be

$$\hat{F}^{i+1}(t) = \hat{x}^i(t+1) - (A\hat{x}^i(t) + Bu(t)). \quad (4.100)$$

As is obvious from the above equation,  $i$  denotes the index of iteration for the estimate of  $F$ . Now, the question is if  $\hat{F}^{i+1}$  is better than  $\hat{F}^i$ . In order to see this, define the bias estimate error  $\tilde{F}^i(t) = F - \hat{F}^i(t)$ . Since  $F$  is constant, it is enough to consider only the steady state error  $\tilde{F}^i = F - \hat{F}^i$ . Rewrite (4.100) as

$$\hat{x}^i(t+1) = A\hat{x}^i(t) + Bu(t) + \hat{F}^{i+1}(t). \quad (4.101)$$

Hence, the bias estimate error  $\tilde{F}$  can be obtained from (4.101) and (4.94) as

$$\tilde{x}^i(t+1) = A\tilde{x}^i(t) + \tilde{F}^{i+1}(t). \quad (4.102)$$

In steady state,  $\tilde{x}^i(t) = \tilde{x}^i(\infty)$  and we can obtain

$$\tilde{F}^{i+1} = (I - A)\tilde{x}^i(\infty) \quad (4.103)$$

where  $I$  is the identity matrix. We denote  $\tilde{F}^i(\infty)$  as  $\tilde{F}^i$  for convenience. Then from (4.99), we can obtain the error dynamics of the bias estimation as

$$\tilde{F}^{i+1} = (I - A)(I - A + LC)^{-1}\tilde{F}^i. \quad (4.104)$$

We can simplify the equation by matrix inversion Lemma (Brogan, 1991),

$$\begin{aligned} (I - A + LC)^{-1} &= (I - A)^{-1} - (I - A)^{-1}L(I + C(I - A)^{-1}L)^{-1}C(I - A)^{-1} \\ &= (I - A)^{-1} - \alpha(I - A)^{-1}LC(I - A)^{-1} \end{aligned} \quad (4.105)$$

where  $\alpha = 1/(1 + C(I - A)^{-1}L)$ . Then,

$$(I - A)(I - A + LC)^{-1} = I - \alpha LC(I - A)^{-1} \quad (4.106)$$

and the bias error dynamics is simplified as

$$\tilde{F}^{i+1} = [I - \alpha LC(I - A)^{-1}]\tilde{F}^i. \quad (4.107)$$

(4.107) shows that  $\tilde{F}^i \rightarrow 0$  as  $i \rightarrow \infty$  if all the eigenvalues of  $I - \alpha LC(I - A)^{-1}$  is inside unit circle. However, this is not the case as can be shown in the following proposition.

**Proposition 4.6 (Convergence of iterative bias estimation using Luenberger observer)** Given the simplified CASS model in (4.94), the Luenberger observer based iterative bias estimation does not converge to the true bias, that is,

$$\tilde{F}^\infty \neq 0. \quad (4.108)$$

**Proof:** Consider the steady state of  $\tilde{F}^i$ , that is,  $\tilde{F}^i = \tilde{F}^\infty$ , then from (4.107), we can see that

$$\tilde{F}^\infty = (I - \alpha LC(I - A)^{-1})\tilde{F}^\infty \quad (4.109)$$

$$= \tilde{F}^\infty - \alpha LC(I - A)^{-1}\tilde{F}^\infty, \quad (4.110)$$

which results in

$$\alpha LC(I - A)^{-1}\tilde{F}^\infty = 0. \quad (4.111)$$

However,  $LC(I - A)^{-1}$  is rank 1 because

$$rk(LC(I - A)^{-1}) \leq \min(rk(LC), rk(I - A)^{-1}) \quad (4.112)$$

and  $rk(LC) = 1$ . Therefore, we can conclude that  $\tilde{F}^\infty$  has nonzero steady state solution.  $\square$

Proposition 4.6 shows that Luenberger observer based iterative method does not provide the unbiased estimate of  $F$ . This is somewhat intuitive since Luenberger observer does not explicitly deal with disturbances, and  $\tilde{F}^i$  in (4.98) can be interpreted as bounded disturbance.

Now, we might wonder if we can find a remedy to make the bias estimation dynamics (4.107) stable. Reconsider (4.107) and place some extra variable to stabilize the dynamics as

$$\tilde{F}^{i+1} = A_F \tilde{F}^i + v^i \quad (4.113)$$

where  $A_F = [I - \alpha LC(I - A)^{-1}]$  and  $v^i$  is an extra variable to stabilize the error dynamics of the bias estimation. (4.113) appears to be a typical state feedback

control problem, however, the state feedback  $\tilde{F}^i$  is not available since  $F$  is not known. Instead, try  $v^i = K\hat{F}^i$ . This additional term is same as modifying observer as

$$\hat{x}^i(t+1) = A\hat{x}^i(t) + Bu(t) + \hat{F}^i + L(y(t) - \hat{y}^i(t)) - K\hat{F}^i \quad (4.114)$$

$$= A\hat{x}^i(t) + Bu(t) + (I - K)\hat{F}^i + L(y(t) - \hat{y}^i(t)) \quad (4.115)$$

$$\hat{y}^i(t) = C\hat{x}^i(t). \quad (4.116)$$

That is, the state offset estimate is multiplied by non-unity gain. Then, the new error dynamics becomes

$$\tilde{F}^{i+1} = A_F\tilde{F}^i + K\hat{F}^i \quad (4.117)$$

$$= (A_F - K)\tilde{F}^i + KF \quad (4.118)$$

by adding and subtracting  $KF$  in the right hand side. We may decide  $K$  using any linear controller design method such as LQR or pole placement. Then in steady state, we obtain

$$(I - A_F + K)\tilde{F}^\infty = KF. \quad (4.119)$$

Since  $\tilde{F}^\infty = F - \hat{F}^\infty$ , the above equation can be simplified as

$$(I - (I - A_F + K)^{-1}K)F = \hat{F}^\infty. \quad (4.120)$$

So,  $F$  can be recovered from  $\hat{F}^\infty$  if the matrix in the left hand side is invertible. Unfortunately, this is not the case, either. Since  $A_F = [I - \alpha LC(I - A)^{-1}]$ ,  $I - A_F + K$  is

$$I - A_F + K = I - [I - \alpha LC(I - A)^{-1}] + K \quad (4.121)$$

$$= \alpha LC(I - A)^{-1} + K. \quad (4.122)$$

With the above equation, (4.119) becomes

$$(\alpha LC(I - A)^{-1} + K)(F - \hat{F}^\infty) = KF, \quad (4.123)$$

then

$$(\alpha LC(I - A)^{-1} + K)\hat{F}^\infty = \alpha LC(I - A)^{-1}F. \quad (4.124)$$

Since the matrix in the right hand side of (4.124) is not invertible because  $LC(I - A)^{-1}$  is rank 1,  $F$  cannot be recovered from  $\hat{F}^\infty$ .

The above results show that Luenberger observer based method cannot recover the true state biases. Now, the question is ‘is there any observer that can estimate the unknown states correctly in the presence of step disturbance?’ For this purpose, we investigate robust observers.

Sliding mode control is a robust variable structure control which can ensure performance and stability in the presence of bounded uncertainties. Motivated from the robustness of sliding mode control, we try to find the counterpart in the observer area. Since sliding mode observer is closely related to sliding mode control, we present two discrete-time sliding mode controller in Appendix B. Despite the similarity between sliding mode controller and observers, they have differences in that sliding mode controller is based on state feedback while only output feedback is used in sliding mode observer. Because only the output feedback is available in observer design by its nature, the uncertainty of the system in sliding mode observer design is usually limited to the canonical form, that is,

$$\begin{aligned} x_1(t+1) &= x_2(t) \\ &\vdots \\ x_n(t+1) &= g(x(t), u(t)) \end{aligned}$$

where  $g(\cdot)$  is a nonlinear/uncertainty function of state, input, and disturbances (Choi et al., 1999). Also, the literature is scarce about discrete time system with general uncertainty. The discrete time sliding mode observer reported in (Haskara and Utkin, 1998) is only for nominal model. In the rest of this subsection, we derive a discrete-time sliding mode observer, which is an extension of the discrete-time sliding observer

proposed in (Haskara and Utkin, 1998) by accounting for the bounded model uncertainty. The second controller in Appendix B is used for the purpose, because of its simplicity compared with the first one.

To start the derivation, reconsider the CASS model in observable form (4.28)-(4.33). Since the model is already in a canonical form, we can rewrite the equation as

$$y(t+1) = A_{11}y(t) + A_{12}x_r(t) + B_1u(t) + F_1 \quad (4.125)$$

$$x_r(t+1) = A_{21}y(t) + A_{22}x_r(t) + B_2u(t) + F_2 \quad (4.126)$$

where  $x_r(t)$  is the remainder states, i.e.  $x_r(t) = [x_2(t), \dots, x_n(t)]^T$ .  $A_{ij}$  is the block matrix of  $A$  matrix of (4.28). That is,

$$\left[ \begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right] = \left[ \begin{array}{c|cccc} a_1 & 1 & 0 & \dots & 0 \\ a_2 & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & 0 & 0 & \dots & 1 \\ a_n & 0 & 0 & \dots & 0 \end{array} \right]. \quad (4.127)$$

$$\left[ F_1 \mid F_2 \right]^T = \left[ f_1 \mid f_2 \quad \dots \quad f_n \right]^T \quad (4.128)$$

$$\left[ B_1 \mid B_2 \right]^T = \left[ b_1 \mid b_2 \quad \dots \quad b_n \right]^T. \quad (4.129)$$

Then, the modified form of the observer proposed in (Haskara and Utkin, 1998) has the following

$$\hat{y}(t+1) = A_{11}\hat{y}(t) + A_{12}\hat{x}_r(t) + B_1u(t) + \hat{F}_1 - v(t) \quad (4.130)$$

$$\hat{x}_r(t+1) = A_{21}\hat{y}(t) + A_{22}\hat{x}_r(t) + B_2u(t) + \hat{F}_2 + Lv(t) \quad (4.131)$$

and the observer error dynamics is obtained from (4.125)-(4.126) and (4.130)-(4.131) as

$$\tilde{y}(t+1) = A_{11}\tilde{y}(t) + A_{12}\tilde{x}_r(t) + \tilde{F}_1 + v(t) \quad (4.132)$$

$$\tilde{x}_r(t+1) = A_{21}\tilde{y}(t) + A_{22}\tilde{x}_r(t) + \tilde{F}_2 - Lv(t). \quad (4.133)$$

The sliding surface for the observer is

$$s(t) = Cx(t), \quad (4.134)$$

that is, the output  $y(t)$ .  $v(t)$  in the observer is an additional term to compensate the error caused from unknown initial conditions and/or model uncertainties. Choose  $v(t)$  as

$$v(t) = v_{eq}(t) - \Delta v(t) \quad (4.135)$$

$v_{eq}(t)$  is the equivalence control term for the case without any model uncertainty, that is, the equivalence controller on the sliding surface. From (4.132), we can obtain  $v_{eq}(t)$  by ignoring  $\tilde{F}_1(t)$ . The resulting  $v_{eq}(t)$  is

$$v_{eq}(t) = -(A_{11}\tilde{y}(t) + A_{12}\tilde{x}_r(t)). \quad (4.136)$$

However,  $\tilde{x}_r(t)$  is not available. Therefore, we have to rely on the estimate of  $\tilde{x}_r(t)$  instead, as

$$v_{eq}(t) = -(A_{11}\tilde{y}(t) + A_{12}\hat{\tilde{x}}_r(t)). \quad (4.137)$$

In order to obtain the estimate of  $\tilde{x}_r(t)$ , replace  $v(t)$  in (4.133) with (4.135). And we obtain

$$\begin{aligned} \tilde{x}_r(t+1) &= A_{21}\tilde{y}(t) + A_{22}\tilde{x}_r(t) + \tilde{F}_2 \\ &\quad - L(-(A_{11}\tilde{y}(t) + A_{12}\tilde{x}_r(t)) - \Delta v(t)) \end{aligned} \quad (4.138)$$

$$= (A_{21} + LA_{11})\tilde{y}(t) + (A_{22} + LA_{12})\tilde{x}_r(t) + \tilde{F}_2 + L\Delta v(t). \quad (4.139)$$

We may obtain the estimate of  $\tilde{x}_r(t)$  from the above equation as

$$\hat{\tilde{x}}_r(t+1) = (A_{21} + LA_{11})\tilde{y}(t) + (A_{22} + LA_{12})\hat{\tilde{x}}_r(t) + L\Delta v(t). \quad (4.140)$$

Define the error of the estimate of  $\tilde{x}_r(t)$  as  $e(t)$ , that is,  $e(t) = \tilde{x}_r(t) - \hat{\tilde{x}}_r(t)$  and by subtracting (4.140) from (4.139), then we obtain

$$e(t+1) = (A_{22} + LA_{12})e(t) + \tilde{F}_2. \quad (4.141)$$

Hence, we can see that the error of state estimation error is bounded if  $L$  is chosen to guarantee stability. The resulting error dynamics of  $\hat{y}(t)$  is

$$\tilde{y}(t+1) = A_{12}e(t) + \tilde{F}_1 - \Delta v(t). \quad (4.142)$$

The remaining state error dynamics is obtained by replacing  $v(t)$  in (4.133) with (4.135) as

$$\begin{aligned} \tilde{x}_r(t+1) &= (A_{21} + LA_{11})\tilde{y}(t) + A_{22}\tilde{x}_r(t) \\ &\quad + LA_{12}\hat{\tilde{x}}_r(t) + \tilde{F}_2 + L\Delta v(t). \end{aligned} \quad (4.143)$$

Since  $e(t) = \tilde{x}_r(t) - \hat{\tilde{x}}_r(t)$ ,  $\hat{\tilde{x}}_r(t) = \tilde{x}_r(t) - e(t)$ . Then, the remaining state error dynamics can be rewritten as

$$\begin{aligned} \tilde{x}_r(t+1) &= (A_{21} + L(A_{11} + a))\tilde{y}(t) + (A_{22} + LA_{12})\tilde{x}_r(t) \\ &\quad + \tilde{F}_2 - LA_{12}e(t). \end{aligned} \quad (4.144)$$

Now, decide  $\Delta v(t)$ . From the second controller in the Appendix B, the sliding mode  $\tilde{y}(t)$  must satisfy the following equation in order to have attracting boundary layers.

$$(\tilde{y}(t+1) - \tilde{y}(t))(\tilde{y}(t+1) + \tilde{y}(t)) < 0. \quad (4.145)$$

Then From (4.142), the above condition becomes

$$(A_{12}e(t) + \tilde{F}_1 - \Delta v(t) - \tilde{y}(t))(A_{12}e(t) + \tilde{F}_1 - \Delta v(t) + \tilde{y}(t)) < 0 \quad (4.146)$$

which is satisfied if

$$A_{12}e(t) + \tilde{F}_1 - \tilde{y}(t) < \Delta v(t) < A_{12}e(t) + \tilde{F}_1 + \tilde{y}(t) \quad \text{if } \tilde{y}(t) > 0 \quad (4.147)$$

$$A_{12}e(t) + \tilde{F}_1 + \tilde{y}(t) < \Delta v(t) < A_{12}e(t) + \tilde{F}_1 - \tilde{y}(t) \quad \text{if } \tilde{y}(t) < 0. \quad (4.148)$$

Because of the definition of block matrix  $A_{12}$ ,  $A_{12}e(t) = e_1(t)$ . It is common to choose the observer gain so that observer has dead beat response. Also, for the simplicity of



the derivation, choose  $L = 0$ , which leads to dead beat response of the error dynamics (4.141). Then, from (4.141), we can see that

$$e_1(\infty) = \sum_2^n \tilde{f}_i. \quad (4.149)$$

Therefore, it is obvious that

$$|A_{12}e(\infty) + \tilde{F}_1| = \left| \sum_1^n \tilde{f}_i \right| < \sum_1^n |\tilde{f}_i|. \quad (4.150)$$

From Appendix B, the boundary layer can be chosen to be

$$\phi = \sum_1^n |\tilde{f}_i| + \epsilon_\phi. \quad (4.151)$$

Therefore, we can choose  $\Delta v(t)$  as

$$\Delta v(t) = \begin{cases} 0 & \text{if } |\tilde{y}(t)| > \phi \\ a\tilde{y}(t) & \text{if } |\tilde{y}(t)| \leq \phi \end{cases} \quad (4.152)$$

where  $\phi$  is a boundary layer and is chosen as  $\phi = \sum_1^n |\tilde{f}_i| + \epsilon_\phi$ . Here,  $\epsilon_\phi$  is a small positive constant.  $a$  must be chosen to be  $0 < a < \epsilon_\phi/\phi$ .

In steady state of  $t \rightarrow \infty$ , the error dynamics are written as

$$(I - A_{22} - LA_{12})e(\infty) = \tilde{F}_2 \quad (4.153)$$

$$(1 + a)\tilde{y}(\infty) = A_{12}e(\infty) + \tilde{F}_1 \quad (4.154)$$

$$(I - A_{22} - LA_{12})\tilde{x}_r(\infty) = (A_{21} + L(A_{11} + a))\tilde{y}(\infty) - LA_{12}e(\infty) + \tilde{F}_2 \quad (4.155)$$

Simplify the error dynamics by choosing  $L = 0$ , the observer gain for dead beat response and compare the sliding mode observer with Luengerber observer with dead beat observer error dynamics in the following proposition.

**Proposition 4.7 (Steady state error of sliding mode observer)** The proposed sliding mode observer has reduced effects of unknown state offsets on the first state  $x_1(t)$  than Luenberger observer.

**Proof:** The steady state error of Luenberger observer is from (4.99),

$$(I - A + LC)\tilde{x}_l = \tilde{F}, \quad (4.156)$$

where subscript  $l$  represent Luenberger observer. We omit  $\infty$  in this proof to simplify the notation. For comparison with sliding mode observer, choose the observer gain so that the error dynamics has dead beat response. The dead beat response gain is  $L_l = [a_1, \dots, a_n]^T$ . Since the system matrices are in canonical observable form, (4.156) with this gain has the following special structure

$$\begin{bmatrix} 1 & -1 & 0 & \dots & \\ 0 & 1 & -1 & 0\dots & \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \tilde{x}_l = \begin{bmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_n \end{bmatrix} \quad (4.157)$$

By adding all the rows, we can single out a solution for  $(\tilde{x}_1)_l$  as

$$(\tilde{x}_1)_l = \sum_1^n \tilde{f}_i. \quad (4.158)$$

We can solve for the rest of the steady state errors and obtain

$$(\tilde{x}_2)_l = \sum_2^n \tilde{f}_i \quad (4.159)$$

$$(\tilde{x}_3)_l = \sum_3^n \tilde{f}_i \quad (4.160)$$

$$\vdots \quad (4.161)$$

$$(\tilde{x}_n)_l = \tilde{f}_n. \quad (4.162)$$

In contrast, sliding mode observer has more complex stead state error equations as (4.153)-(4.155). For notational convenience, we omit subscript  $s$  for the steady state errors of sliding mode observer. Similarly, (4.153) has with  $L = 0$  for dead beat

response leads to the solution,

$$e_1 = \sum_2^n \tilde{f}_i \quad (4.163)$$

$$e_2 = \sum_3^n \tilde{f}_i \quad (4.164)$$

$$e_3 = \sum_4^n \tilde{f}_i \quad (4.165)$$

$$\vdots \quad (4.166)$$

$$e_{n-1} = \tilde{f}_n. \quad (4.167)$$

where  $e = [e_1, \dots, e_{n-1}]^T$ . From (4.154) and (4.163), we obtain

$$(1+a)\tilde{y} = e_1 + \tilde{f}_1 \quad (4.168)$$

which is simplified as

$$\tilde{y} = \frac{1}{1+a} \sum_1^n \tilde{f}_i \quad (4.169)$$

Compare (4.158) with (4.169) and we can see that  $e_1$  is smaller than  $(e_1)_l$  since  $1+a > 1$ .

The rest of state observer errors can be solved from (4.155). Similar with (4.163)-(4.167),

$$\tilde{x}_2 = \sum_2^n a_i \left( \sum_1^n \tilde{f}_j \frac{1}{1+a} \right) + \sum_2^n \tilde{f}_i \quad (4.170)$$

$$\tilde{x}_3 = \sum_3^n a_i \left( \sum_1^n \tilde{f}_j \frac{1}{1+a} \right) + \sum_3^n \tilde{f}_i \quad (4.171)$$

$$\vdots \quad (4.172)$$

$$\tilde{x}_n = a_n \left( \sum_1^n \tilde{f}_j \frac{1}{1+a} \right) + \tilde{f}_n. \quad (4.173)$$

Hence, even though the first observed state of sliding mode observer has reduced effects of unknown state offsets, the effects on the remaining states are inconclusive.

□

**Remark 4.4** *The selection of observer gains to be dead beat response is for convenience. However, it is still true for general observer gains that the first states has improved estimates. To see this, take the ratio between  $(\tilde{x}_1)_1$  and  $\tilde{x}_1$  as*

$$\frac{(\tilde{x}_1)_1}{\tilde{x}_1} = (1+a) \frac{\sum_1^n \tilde{f}_i}{\tilde{f}_1(1 - \sum_2^n l_i) + \sum_2^n \tilde{f}_i} \cdot \frac{1 - \sum_2^n l_i}{1 - \sum_1^n a_i + \sum_1^n (l_i)_1}. \quad (4.174)$$

*Since the observer gains have to be chosen to stabilize the observer error dynamics, all the poles of the observer error dynamics must be inside the unit circle, that is, the eigenvalues of (4.141) and (4.98) must be inside the unit circle. This implies that  $1 - \sum_2^n l_i$  and  $1 - \sum_1^n a_i + \sum_1^n (l_i)_1$  are positive since these two equations can be obtained from the characteristic equations by replacing  $z$  with 1. That is,*

$$z^n - l_1 z^{n-1} - \dots - l_{n-1} z - l_n$$

*is the characteristic equation for sliding mode observer. By replacing  $z$  with 1, we obtain  $1 - \sum_2^n l_i$ . Since  $1+a > 1$ , we can see that we can make  $\tilde{x}_1$  reduced by choosing the sliding mode gain properly.*

To the author's knowledge, there is no robust observer that can estimate unbiased states in the presence of uncertainty. The best that we can do is to have bounded response. Similar with Luenberger observer case, we can try iteration of recursive states and bias estimation. The bias estimation error dynamics can be obtained as

$$\tilde{F}^{i+1} = (I - A)\tilde{x}(\infty). \quad (4.175)$$

$\tilde{x}(\infty)$  is a function of  $\tilde{F}^i$ , iteration process may lead to the unbiased estimate of  $F$ . However, numerical computations of the poles of the dynamics with several different systems show that the steady state equation by  $\tilde{F}^\infty$  has only rank one. We investigate the cause of this rank deficiency.

**Proposition 4.8 (Property of observer based iteration for state offset estimation)** Observer based iteration process for bias estimation does not converge to the unbiased estimate.

**Proof:** Consider the original system and an observer

$$x(t+1) = Ax(t) + Bu(t) + F \quad (4.176)$$

$$\hat{x}^i(t+1) = A\hat{x}^i(t) + Bu(t) + \hat{F}^i + w(t)^i \quad (4.177)$$

where  $w(t)^i$  is the additional terms in observer to compensate the model errors and unknown initial conditions. We can easily obtain observer dynamics by subtracting the first equation from the second, then

$$\tilde{x}^i(t+1) = A\tilde{x}^i(t) + \tilde{F}^i - w(t)^i \quad (4.178)$$

The new update of  $\hat{F}^{i+1}$  can be represented as

$$\hat{x}^i(t+1) = A\hat{x}^i(t) + Bu(t) + \hat{F}^{i+1} \quad (4.179)$$

then the error dynamics can be obtained in a similar way

$$\tilde{x}^i(t+1) = A\tilde{x}^i(t) + \tilde{F}^{i+1} \quad (4.180)$$

Subtract (4.178) from (4.179), we obtain

$$\tilde{F}^{i+1}(t) = \tilde{F}^i(t) - w^i(t) \quad (4.181)$$

Suppose that there is steady state in terms of  $i$ , the  $\tilde{F}^i = \tilde{F}^{i+1}$  since  $\tilde{F}^i = \tilde{F}^\infty$ . Hence it implies that  $w(t)^i = 0$  in steady state, which shows that additional control from observer does not have influence in  $\tilde{F}^i$  in steady state as  $i \rightarrow \infty$ .  $\square$

**Remark 4.5** For Luenberger observer,  $w^i(t) = L\tilde{y}^i(t)$ .  $L$  is  $n \times 1$  vector, which shows that  $\tilde{y}^i$  will have infinite number of solutions. For the sliding mode observer,

$$w^i(t) = \begin{bmatrix} -1 \\ L \end{bmatrix} v(t). \quad (4.182)$$

## 4.5 Discussion

In this chapter, we investigated direct identification of CASS model. Motivations of studying this type of model structures are illustrated. Even though the need of CASS model for control is obvious, the identification of the model is not trivial and revealed serious problems. Several methods from estimation and control have been investigated but all failed to recover the system. The question that we may wonder is “does it mean that multimodel approach cannot be applied in practice in a rigorous way?” We try to answer this question in the next chapter.

# Chapter 5

## Identification of a CASS model by nonlinear state realization

### 5.1 Introduction

In the previous discussions, we show the limitations of TS fuzzy models from system identification point of view. An alternative model structure, so called CASS model was proposed, because of the problems of conventional recursive i/o type local models. However, this local model structure is turned out to be unidentifiable despite the clear interpretation in terms of dynamic systems and the obvious demands from multimodel based control. The significance of unidentifiability was analyzed in several different model structures and estimation methods. For the original form of CASS model, the unidentifiability is caused because only the output is assumed to be measured. Hence, we can expect that the unidentifiability of the CASS model can be relaxed with full state measurement. Motivated from this possibility of CASS model identification with full state measurement, we attempt a solution from totally different perspective, which is to utilize state realization of intermediate nonlinear i/o model. The i/o model is only intermediate since our ultimate goal is to identify CASS model for multimodel composition.

Our approach can also be considered as an alternative to use nonlinear black box models. In intelligent control literatures, controllers utilizing NARX type nonlinear

models are usually trained through the model instead of being designed. This is because the complexity as well as nonlinearity of the model make any analytical design approach of controller infeasible, in spite of the absence of analytical assurance of nominal or robust stability and performance. Our contributions in this chapter is, therefore, can be summarized as the proposal of nonconventional way to handle the unidentifiability of CASS model as well as proposing a way for the design of controllers using black box nonlinear i/o models.

This chapter is not a complete chapter. Rather, the purpose of this chapter is to introduce a new idea and identify problems for future research. Since data based state realization is still immature and there is no working solution for our needs, this indirect identification method is premature to be fully addressed at this moment.

This chapter is organized as follows. We propose a data based state realization method for linear systems and utilize the state estimate for the identification of state space model in Section 5.2. In Section 5.3, we review nonlinear state realization, emphasized on data based state realization. In Section 5.4, recursive i/o model for state realization is investigated. We address issues of state realization for CASS model identification in Section 5.5. A simple example is used to illustrate the ideas and problems yet to be solved in Section 5.6.

## **5.2 Data based state realization of linear system**

For linear systems, state realization is usually obtained from transforming i/o equation into transfer functions (Brogan, 1991). Even though this realization is straightforward, it has problems in applying to data based state realization. In this Chapter, we present a state realization technique that can be applied to data based state realization in a straightforward manner. This realization is different from Nerode realization using impulse response as in (Leontaritis and Billings, 1985) and is not found in any text.



Consider a linear state space system in observable form:

$$x(t+1) = Ax(t) + Bu(t) \quad (5.1)$$

$$y(t) = Cx(t) \quad (5.2)$$

and the linear difference i/o equation

$$y(t) + a_1y(t-1) + \dots + a_ny(t-n) = b_1u(t-1) + \dots + b_nu(t-n), \quad (5.3)$$

and state realization is to transform the difference i/o equation into state space system and vice versa. Firstly, consider obtaining state space from the i/o equation.

Let

$$x_1(t) = y(t) \quad (5.4)$$

and from (5.3),

$$y(t) = -(a_1y(t-1) + \dots + a_ny(t-n)) + (b_1u(t-1) + \dots + b_nu(t-n)). \quad (5.5)$$

Shift the time index  $t$  by 1 and we obtain the state equation for the first state as

$$x_1(t+1) = y(t+1) \quad (5.6)$$

$$\begin{aligned} &= -(a_2y(t-2) + \dots + a_ny(t-n)) \\ &\quad + (b_2u(t-1) + \dots + b_nu(t-n)) - a_1y(t) + b_1u(t) \end{aligned} \quad (5.7)$$

$$\begin{aligned} &= -\{(a_2y(t-2) + \dots + a_ny(t-n)) \\ &\quad + (b_2u(t-1) + \dots + b_nu(t-n))\} - a_1x_1(t) + b_1u(t). \end{aligned} \quad (5.8)$$

Since state equation is only a function of states by its definition, we denote the equation inside  $\{\}$  as  $x_2(t)$ . Increase the index of  $t$  in  $x_2(t)$  by 1 and follow the same procedure, then we obtain

$$\begin{aligned} x_2(t+1) &= -(a_3y(t-3) + \dots + a_ny(t-n)) \\ &\quad + (b_3u(t-1) + \dots + b_nu(t-n)) - a_2y(t) + b_2u(t) \end{aligned} \quad (5.9)$$

$$= x_3(t) - a_2x_1(t) + b_2u(t). \quad (5.10)$$

Continue this procedure until we exhaust all the  $y(t)$  and  $u(t)$  terms as

$$\begin{aligned} x_{n-1}(t+1) &= -(a_n y(t-1) + b_n u(t-n)) \\ &\quad - a_{n-1} y(t) + b_{n-1} u(t) \end{aligned} \quad (5.11)$$

$$= x_n(t) - a_{n-1} x_1(t) + b_{n-1} u(t) \quad (5.12)$$

$$x_n(t+1) = -a_n y(t) + b_n u(t) \quad (5.13)$$

$$= -a_n x_1(t) + b_n u(t). \quad (5.14)$$

Rewrite the set of equations and we obtain the usual observable canonical state space realization,

$$x(t+1) = \begin{bmatrix} -a_1 & 1 & 0 & \dots & \dots & 0 \\ -a_2 & 0 & 1 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ -a_n & 0 & 0 & \dots & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} u(t) \quad (5.15)$$

$$y(t) = [1 \ 0 \ \dots \ 0] x(t) \quad (5.16)$$

From the above procedure, we can deduce the state realization based on i/o data rather than i/o equation. That is,

$$\begin{aligned} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} &= \begin{bmatrix} -a_n & -a_{n-1} & \dots & -a_2 & -a_1 \\ 0 & -a_n & \dots & -a_3 & -a_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & -a_n \end{bmatrix} \begin{bmatrix} y(t-n) \\ y(t-n+1) \\ \vdots \\ y(t-1) \end{bmatrix} \\ &+ \begin{bmatrix} b_n & b_{n-1} & \dots & b_2 & b_1 \\ 0 & b_n & \dots & b_3 & b_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & b_n \end{bmatrix} \begin{bmatrix} u(t-n) \\ u(t-n+1) \\ \vdots \\ u(t-1) \end{bmatrix}. \end{aligned} \quad (5.17)$$

Note that dimension of state space is same as the order of the regression or the maximum lag of the i/o's. This implies that the state space that is obtained through this procedure is not necessarily minimal. Since minimality of state space cannot be verified without the state space equations, we have to identify the state space equation from the estimated states and the controllability and observability of the state space equations must be checked. The identification of state space equation from estimated

states is a simple linear least squares problem. We illustrate this idea in the following example.

**Example 5.1** Consider an arbitrary 3rd order stable linear discrete-time system

$$x(t+1) = Ax(t) + Bu(t) \quad (5.18)$$

$$y(t) = Cx(t) + Du(t) \quad (5.19)$$

with system matrices as

$$A = \begin{bmatrix} 0.18558 & 0.1237 & 0.35108 \\ 0.2992 & 0.23246 & -0.18388 \\ -0.22144 & 0.32868 & 0.018463 \end{bmatrix} \quad (5.20)$$

$$B = [-0.85588 \ 0 \ 0]^T \quad (5.21)$$

$$C = [-0.98658 \ 0 \ 0.18806] \quad (5.22)$$

$$D = 0. \quad (5.23)$$

If  $x(t)$  is measured, this becomes a straightforward linear least squares problem as

$$Y = \Phi\theta \quad (5.24)$$

where the row of  $\Phi$  and  $Y$  are  $[x^T(t), u^T(t)]$  and  $[x^T(t+1), y(t)]$ , respectively. Here  $\theta$  is

$$\theta = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^T. \quad (5.25)$$

The state  $x(t)$  can be estimated from (5.17) once the intermediate i/o equation is estimated. The identification of i/o equation can be done by linear least squares. Generate 1000 points of i/o data with input  $\mathcal{N}(0, 1)$ . The selection of system order is quite straightforward since the least squares for the identification of i/o equation becomes singular if the order of regression exceeds 3. Therefore, the i/o model with nonsingular least squares solution can be considered as the minimal order of the system. The

estimated i/o equation with order 3 accurately recovered the transfer function of the system since there is no noise in the system. The identified i/o equation is

$$y(t) = 0.4365y(t-1) - 0.1520y(t-2) + 0.0690y(t-3) \\ + 0.8444u(t-1) - 0.1762u(t-2) + 0.0305u(t-3). \quad (5.26)$$

Since initial conditions are not known, it is better to use simulated outputs for the state estimation. Simulate the i/o equation and the state estimate can be obtained from

$$x_1(t) = \hat{y}(t) \quad (5.27)$$

$$x_2(t) = -0.152\hat{y}(t-1) + 0.069\hat{y}(t-2) \\ - 0.176u(t-1) + 0.0305u(t-2) \quad (5.28)$$

$$x_3(t) = 0.0690\hat{y}(t-1) + 0.0305u(t-1). \quad (5.29)$$

This state estimates is the minimal observable state realization and we can obtain  $\hat{A}$  and  $\hat{B}$  from this state estimate using linear least squares. As expected, the matrices are recovered exactly as

$$\hat{A} = \begin{bmatrix} 0.43649 & 1 & -3.8698 \times 10^{-15} \\ -0.15202 & 1.0608 \times 10^{-16} & 1 \\ 0.068963 & -2.7418 \times 10^{-16} & -2.8035 \times 10^{-16} \end{bmatrix} \quad (5.30)$$

$$\hat{B} = [ 0.84439 \quad -0.17623 \quad 0.030542 ]^T. \quad (5.31)$$

Here the terms with  $10^{-16}$  is caused by numerical error and can be considered as zero's. Compare this matrix with (5.15)-(5.16) and we can see that the linear state space is accurately identified.

### 5.3 State realization of nonlinear recursive i/o difference equations

Inspired from the data based linear state realization and the use for the identification of state space equation, it is natural to seek the extension in nonlinear systems.

However, the state realization of nonlinear system is not as straightforward as the case of linear systems. The complication is caused by complex relationship between recursive i/o and state space equations of nonlinear systems. Also, the recursive definition of states are generally not possible for nonlinear systems since regression variables are not isolated by summation as in linear systems. In this and the following sections, we illustrate the complication of nonlinear state realization.

For nonlinear systems, state realization is to construct a state equation

$$x(t+1) = f(x(t), u(t)) \quad (5.32)$$

$$y(t) = h(x(t)) \quad (5.33)$$

for the i/o difference equation

$$y(t) = g(y_{t-n_y}^{t-1}, u_{t-n_u}^{t-1}) \quad (5.34)$$

such that the sequence  $(u(t), y(t))$  generated by (5.32)-(5.33) are same as the ones from (5.34) with zero initial conditions, i.e. they are equivalent. Here,  $y_{t-n_y}^{t-1} = [y(t-1), \dots, y(t-n_y)]^T$  and  $u_{t-n_u}^{t-1} = [u(t-1), \dots, u(t-n_u)]^T$ . Recently, the importance of state realization of recursive i/o difference models (5.34) has been recognized because of the gap between nonlinear system identification using recursive i/o equations and controller design based on the description of (5.32)-(5.33) (Sadegh, 2001).

Given recursive i/o equation like (5.34), one trivial state realization can be obtained by defining state variables as  $x(t) = [y(t), \dots, y(t-n_y+1), u(t-1), \dots, u(t-$

$n_u + 1)]^T$  then

$$\begin{aligned}
 x_1(t+1) &= g(x_1(t), \dots, x_{n_y}(t), x_{n_y+1}(t), \dots, x_{n_y+n_u-1}(t), u(t)) \\
 x_2(t+1) &= x_1(t) \\
 &\vdots \\
 x_{n_y}(t+1) &= x_{n_y-1}(t) \\
 x_{n_y+1}(t+1) &= u(t) \\
 x_{n_y+2}(t+1) &= x_{n_y+1}(t) \\
 &\vdots \\
 x_{n_y+n_u-1}(t+1) &= x_{n_y+n_u-2}(t)
 \end{aligned}$$

and the output relation is

$$y(t) = x_1(t).$$

However, the states of this realization are redundant. For control purpose, minimal realization is important. Non-redundant or minimal realization is, in general, defined such that all the states are both observable and reachable (Sontag, 1979), while more strong conditions can be used such as controllability and observability of the linearized systems at every linearization point (Sadegh, 2001). In addition to the control purpose, minimality of state realization is important for system identification in state space, because minimal state realization is unique in the sense that two different minimal realizations are isomorphic, that is, one-to-one correspondence. In linear systems, the isomorphism is reduced to the mapping by similarity transformation matrix (Sontag, 1979).

Most of state realizations in discrete-time domain are focused on Volterra series or polynomial type i/o relationships (Sontag, 1979). In (Diaz and Desrochers, 1988), state affine state realization from (Sontag, 1979) is used to obtain state space model for recursive i/o model instead of i/o model. Kotta et al. (2001) formulated the

state realization problem as complete integrability of differential forms. Minimal realization is guaranteed if the i/o equation is irreducible. However, the realization in these works are based on the manipulation of i/o equations and real application of this state realization to general nonlinear black box model structures is not feasible because of the need of complex mathematical manipulations.

More proper realization for our purpose can be found in (Sadegh, 2001). The idea of state realization reported by (Sadegh, 2001) is to consider a state space form such as (5.32)-(5.33) as the original system then derive a necessary condition for the state realization by utilizing collected i/o equations such as

$$y(t) = h(x(t)) \quad (5.35)$$

$$y(t+1) = h \circ f(x(t), u(t)) \quad (5.36)$$

$$\vdots \quad (5.37)$$

$$y(t+m-1) = h \circ f^{m-1}(x(t), u_{t+m-2}^t) \quad (5.38)$$

where  $f^i(x(t), u_{t+m}^t) = f_{u(t+m)} \circ f_{u(t+m-1)} \dots \circ f_{u(t)}(x(t))$  with function composition  $\circ$ . Here, the notation of  $f((\cdot), u(t)) = f_{u(t)}(\cdot)$  is used for convenience. Suppose that the state space is not necessarily minimal but observable with  $x(t) \in X \subset \mathbb{R}^m$ . Denote the collected i/o equation in the above by

$$y_{t+m-1}^t = H_m(x(t), u_{t+m-2}^t) \quad (5.39)$$

where  $H_m$  is the vector function with  $h \circ f^{i-1}$  as its component. Then by an application of implicit function theorem, we can obtain  $x(t)$  as an explicit function as

$$x(t) = x(y_{t+m-1}^t, u_{t+m-2}^t). \quad (5.40)$$

Here,  $x(y_{t+m-1}^t, u_{t+m-2}^t)$  denotes that  $x(t)$  is a function of  $y_{t+m-1}^t$  and  $u_{t+m-2}^t$  by some abuse of notation. The application of implicit function theorem is possible because of the imposed observability condition. Now, we can obtain another state equation

as

$$x(t+m) = f^m(x(t), u_{t+m-1}^t) \quad (5.41)$$

$$= f^m(x(y_{t+m-1}^t, u_{t+m-2}^t), u_{t+m-1}^t) \quad (5.42)$$

$$= x(y_{t+m-1}^t, u_{t+m-1}^t). \quad (5.43)$$

Assuming shift-invariance, we can obtain

$$x(t) = x(y_{t-1}^{t-m}, u_{t-1}^{t-m}). \quad (5.44)$$

By replacing  $x(t)$  in (5.33) with (5.44), we obtain a recursive i/o representation from state space equation as

$$y(t) = h(x(y_{t-1}^{t-m}, u_{t-1}^{t-m})). \quad (5.45)$$

Because the implicit function theorem is only valid locally, the obtained recursive i/o equation is valid only locally. Collect i/o equations as

$$y(t) = h(x(y_{t-1}^{t-m}, u_{t-1}^{t-m})) \quad (5.46)$$

$$y(t+1) = h \circ f(x(y_{t-1}^{t-m}, u_{t-1}^{t-m}), u(t)) \quad (5.47)$$

$$\vdots \quad (5.48)$$

$$y(t+m-1) = h \circ f^{m-1}(x(y_{t-1}^{t-m}, u_{t-1}^{t-m}), u_{t+m-2}^t). \quad (5.49)$$

Denote the above collected i/o equation as

$$y_{t+m-1}^t = H_m(x(y_{t-1}^{t-m}, u_{t-1}^{t-m}), u_{t+m-2}^t). \quad (5.50)$$

From (5.50), we can see that (5.50) satisfies the following property

$$\left(D_{y_{t-1}^{t-m}} y_{t+m-1}^t\right)^{-1} D_{u_{t-1}^{t-m}} y_{t+m-1}^t = \left(D_{y_{t-1}^{t-m}} x(t)\right)^{-1} D_{u_{t-1}^{t-m}} x(t) \quad (5.51)$$

where  $D_z x$  is the Jacobian matrix, i.e.  $\frac{\partial x}{\partial z}$ . This can be interpreted that the above equation is independent of  $u_{t+m-1}^t$ . This property can be used as a necessary condition for a state realization.



The state realization from recursive i/o is achieved by the following. Consider a recursive i/o equation

$$y(t) = g(y_{t-1}^{t-m}, u_{t-1}^{t-m}). \quad (5.52)$$

Increase the time index by 1 and we obtain

$$y(t+1) = g(y_t^{t-m+1}, u_t^{t-m+1}) \quad (5.53)$$

$$= g(y_{t-1}^{t-m+1}; y(t), u_{t-1}^{t-m+2}; u(t)). \quad (5.54)$$

Since (5.52) is recursive itself, replace  $y(t)$  in the above equation with (5.52) and define the structure as

$$y(t+1) = g(y_{t-1}^{t-m+1}; g(y_{t-1}^{t-m}, u_{t-1}^{t-m}), u_{t-1}^{t-m+2}; u(t)) \quad (5.55)$$

$$= g_1(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u(t)) \quad (5.56)$$

where  $g_1(\cdot)$  is a function that is originated from  $g(\cdot)$  by replacing  $y(t)$  with  $g(\cdot)$ . We can obtain a collected recursive i/o from this procedure as

$$y(t) = g(y_{t-1}^{t-m}, u_{t-1}^{t-m}) \quad (5.57)$$

$$y(t+1) = g_1(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u(t)) \quad (5.58)$$

$$\vdots \quad (5.59)$$

$$y(t+m-1) = g_{m-1}(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u_{t+m-2}^t) \quad (5.60)$$

where  $g_i(\cdot)$  is a set of functions originated by replacing  $y(t)$  with  $g(\cdot)$  recursively. Denote the above collected i/o equation as

$$y_{t+m-1}^t = G_m(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u_{t+m-2}^t). \quad (5.61)$$

Consider a state vector candidate as

$$x(t) = G_m(y_{t-1}^{t-m}, u_{t-1}^{t-m}, 0) \quad (5.62)$$

where  $G_m(\cdot)$  is a vector function that consist of  $g_i$ . For nonlinear systems, this choice may be contributed from (5.44) which is independent of  $u_{t+m-2}^t$ . Increment the time index  $t$  of (5.62) by one and we obtain

$$x(t+1) = G_m(y_t^{t-m+1}, u_t^{t-m+1}, 0) \quad (5.63)$$

$$= G_m(y_{t-1}^{t-m+1}; y(t), u_{t-1}^{t-m+1}; u(t), 0) \quad (5.64)$$

$$= G_m(y_{t-1}^{t-m+1}; g(y_{t-1}^{t-m}, u_{t-1}^{t-m}), u_{t-1}^{t-m+1}; u(t), 0) \quad (5.65)$$

$$= G_{m+}(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u(t), 0) \quad (5.66)$$

$$= G_{m+}(G_m^{-1}(x(t), u_{t-1}^{t-m}, 0), u_{t-1}^{t-m}, u(t), 0) \quad (5.67)$$

where  $G_m^{-1}(x(t), u_{t-1}^{t-m}, 0)$  is a function that satisfies  $G_m(G_m^{-1}(x(t), u_{t-1}^{t-m}, 0), u_{t-1}^{t-m}, 0) = x(t)$ . This is obtained from the application of implicit function theorem to (5.62). Here,  $G_{m+} = [g_1(\cdot), \dots, g_m(\cdot)]^T$ . If the r.h.s of (5.67) is independent of  $u_{t-1}^{t-m}$ , then the equation is the conventional state realization with output equation

$$y(t) = x_1(t). \quad (5.68)$$

Sadegh (2001) proved that (5.67) is a observable state realization if and only if (5.51) is satisfied. However, (5.51) is quite difficult to verify since it involves differentiation of block matrices and an inverse of one of them. For that reason, he also developed simpler formula to check the existence of observable state realization. Foley and Sadegh (2001) applied this result to several recursive i/o models, however, there is no result on the actual state realization of black box identified model, probably because of the complexity of the regression models.

## 5.4 Identification of recursive i/o model for state realization

As made clear from the previous section, the conventional state realization only exists for limited class of i/o models that satisfy certain properties. As shown in previous

chapters, our main objective in this work is to identify the local state space. Since we assume that data generating system is in state space, the necessary condition for the existence of state space must be satisfied.

The condition (5.51) is quite a general condition that only requires the invertibility of Jacobian of the collected output equations. However, the direct application of the condition to the design of recursive i/o model structure is difficult. Because of the difficulty, Sadegh (2001) also simplified the condition by taking advantage of the recursive nature of the collected i/o equation and cancellation of common matrices in the two Jacobian. The simplified condition is that  $M(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u_{t+m-1}^t)$  is independent of  $u_{t+m-1}^t$ .  $M(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u_{t+m-1}^t)$  is defined as

$$M(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u_{t+m-1}^t) = L_\alpha(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u_{t+m-1}^t)^{-1} L_\beta(y_{t-1}^{t-m}, u_{t-1}^{t-m}, u_{t+m-1}^t) \quad (5.69)$$

where

$$L_\alpha = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,m} \\ 0 & \alpha_{2,1} & \dots & \alpha_{2,m-1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha_{m,1} \end{bmatrix} \quad (5.70)$$

and

$$L_\beta = \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,m} \\ 0 & \beta_{2,1} & \dots & \beta_{2,m-1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \beta_{m,1} \end{bmatrix}. \quad (5.71)$$

Here,  $\alpha_{i,j}$  is the derivative of  $y(t+i)$  with respect to  $y(t-j)$  and  $\beta_{i,j}$  is the derivative of  $y(t+i)$  with respect to  $u(t-j)$ .

Since this is also the sufficient condition for the existence of state realization based on a specific definition of states from collected outputs in (5.62), we can design a regression model structure to guarantee the existence of state realization. In (Sadegh, 2001), a model structure is proposed to satisfy the above condition such as

$$y(t) = \sum_{i=1}^{m-1} g_i(y(t-m+i-1), y(t-m+i), u(t-m+i-1)) + g_m(y(t-1), u(t-1)) \quad (5.72)$$

where  $g_i(\cdot)$  is a function of the variables inside the parenthesis. As can be seen, this is a restricted structure of general nonlinear black box models since the regression variables are not fully connected inside the nonlinear model. We can see that the above regression model satisfies the existence condition of observable state realization. Closer look at (5.72) reveals that this model structure is a limited nonlinear model structure that allows recursive definition of states just like linear state realization.

However, even with this reduced model structure, state realization of the regression model based on the manipulation of equations is still prohibitive. To overcome the difficulty of equation based nonlinear state realization, we investigate data based state realization method using CASS model in the next section.

## 5.5 CASS model identification using the state realization

With the availability of state information, identification of CASS model is a trivial linear squares problem as

$$x(t+1) = [ A \ B \ F ] \begin{bmatrix} x(t) \\ u(t) \\ 1 \end{bmatrix}. \quad (5.73)$$

The problem is how to obtain minimal state realization such that the resulting CASS model is minimal. Since the state realization reviewed in the previous section is only observable instead of minimal, utilization of the state realization may be problematic for the identification of minimal CASS model. In (Sadegh, 2001), an algorithm is proposed to reduce the observable state realization to minimal by selecting only the controllability subspace. However, this reduction process requires to compute the controllability subspaces as well as nonlinear mapping from minimal state space to observable state space. Since the minimal realization is a difficult job, it is attractive if we can use the observable states to identify CASS model then perform the reduction process with the identified CASS model. However, this does not work

as shown in the following.

**Proposition 5.1 (Reduction of CASS model)** The reduction process of CASS model is not same as the reduction process in nonlinear system.

**Proof:** Consider a minimal state realization

$$x(t+1) = f(x(t), u(t)) \quad (5.74)$$

$$y(t) = h(x(t)), \quad (5.75)$$

and an observable state realization

$$z(t+1) = \bar{f}(z(t), u(t)) \quad (5.76)$$

$$y(t) = \bar{h}(z(t)). \quad (5.77)$$

As shown in Lemma 3 in (Sadegh, 2001), there is a smooth map  $\phi$  satisfying

$$z(t) = \phi(x(t)). \quad (5.78)$$

Transformation of CASS model is same as applying (5.78) locally. The local mapping of (5.78) can be written as a linear approximation as

$$z(t) \approx \left. \frac{\partial \phi}{\partial x(t)} \right|_{x_0} (x(t) - x_0) + \phi(x_0). \quad (5.79)$$

Since we do not know  $x_0$ , the affine term cannot be decided and the reduction process is not possible.  $\square$

Because of the reduction of CASS model to minimal does not work locally, it is necessary to have minimal nonlinear state realization initially. However, obtaining nonlinear regression model which leads to minimal realization is quite demanding as explained before. Also, the minimal realization result reported in (Sadegh, 2001) is limited only around zero equilibrium point and cannot be generalized into equilibrium beyond the origin. We illustrate the problem in the following section with an example.

## 5.6 Examples

We reintroduce the nonlinear model in Chapter 3 with modification in the output. For simplicity, we assume a direct measurement of one of the state. Hence, the data generating system is

$$\begin{aligned}\dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= -10\sin(x_1(t)) - 0.5x_2(t) + u(t), \\ y(t) &= x_1(t).\end{aligned}$$

Choose the sampling time  $T = 0.1$  and obtain data by simulating the system with normal  $u(t)$  with mean 7 and unit variance. In this system, the CASS model obtained by linearization and application of numerical integration is a good local state approximation as shown in Fig. 5.1.

We use radial basis functions without bias to identify nonlinear regression model. As suggested in (Sadegh, 2001), specially structured neural networks is used to identify the data generating system in order to have state realization. In this example, we assume that we know the minimal order, in this case, 2, even though this is not feasible in practice. Then, in order to have observable state realization, the regression model with order 2 has the form as

$$y(t) = g_1(y(t-2), y(t-1), u(t-1)) + g_2(y(t-1), u(t-1)). \quad (5.80)$$

where  $g_1(\cdot)$  and  $g_2(\cdot)$  are radial basis functions with different regressors. In Fig. 5.2 and Fig. 5.3, the training and validation results of regression model are given, respectively.

The output shows that regression model identify the local dynamics of nonlinear system fairly well. Estimate the observable states by (5.62). Use the estimated states and estimate CASS model matrices using (5.73). Since the states are not minimal, states are biased as shown in Fig. 5.4

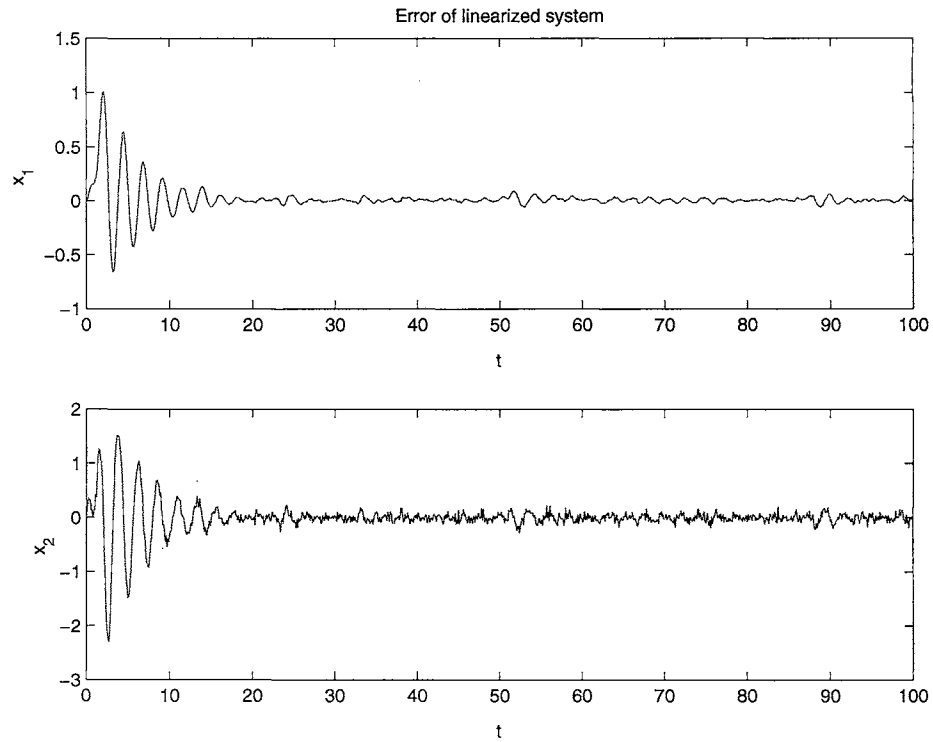


Figure 5.1: Error of CASS model compared with nonlinear system; state 1 (Top figure), state 2 (Bottom figure). The initially large error is because the CASS model is obtained from linearization away from origin. We can observe that CASS model is a good representation of local nonlinear systems as the states of nonlinear system reaches the region that the CASS model is obtained.

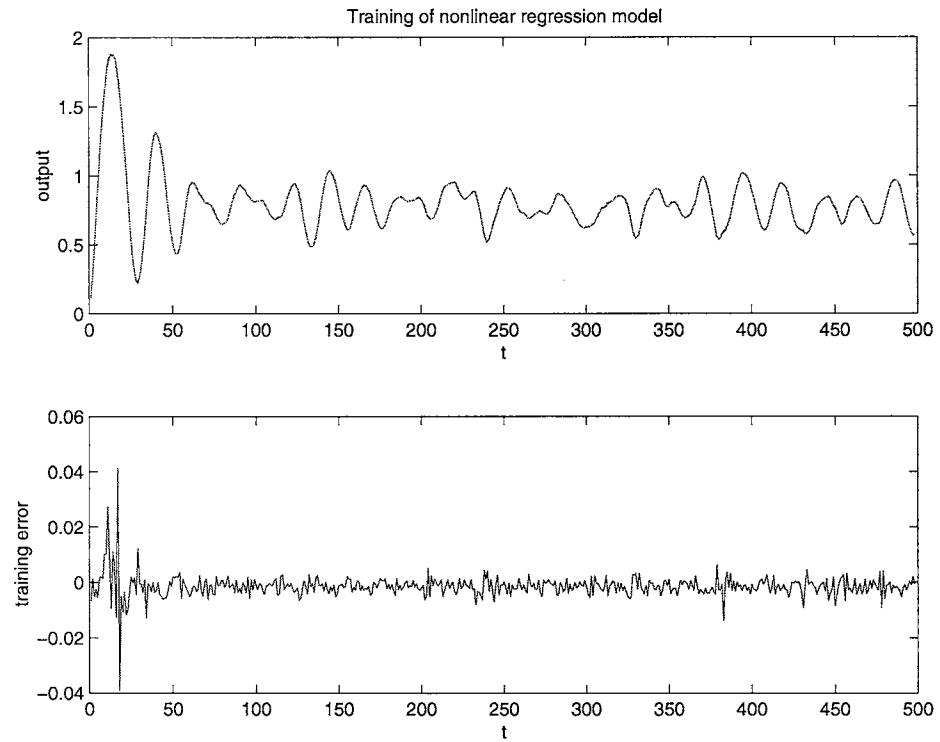


Figure 5.2: Training results of nonlinear regression model; output (Top figure), training error (Bottom figure).



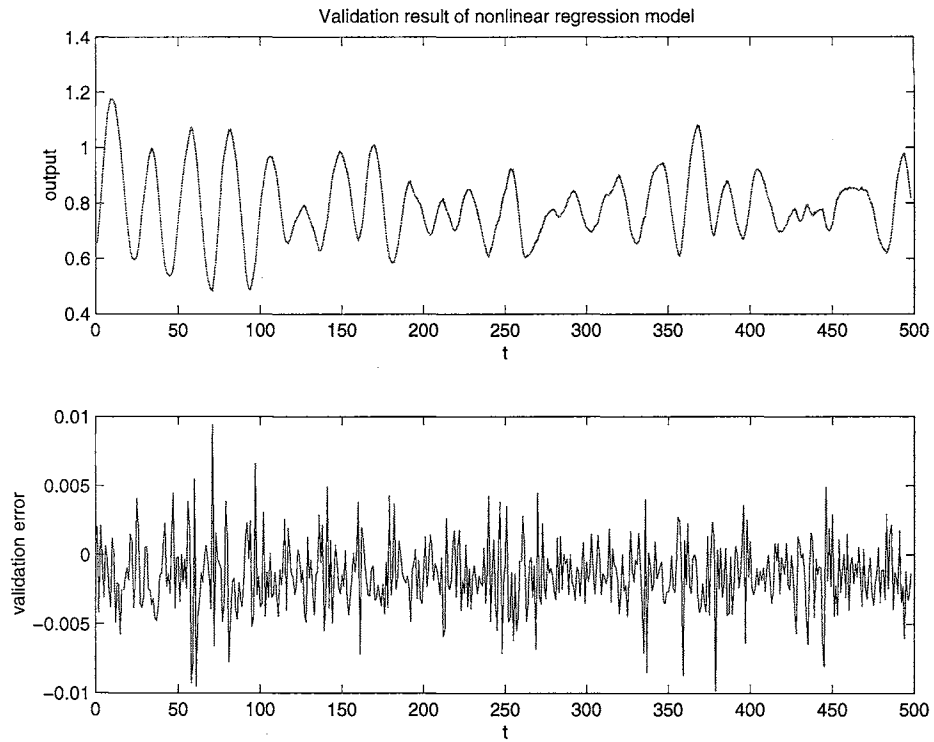


Figure 5.3: Validation results of nonlinear regression model; output (Top figure), validation error (Bottom figure).

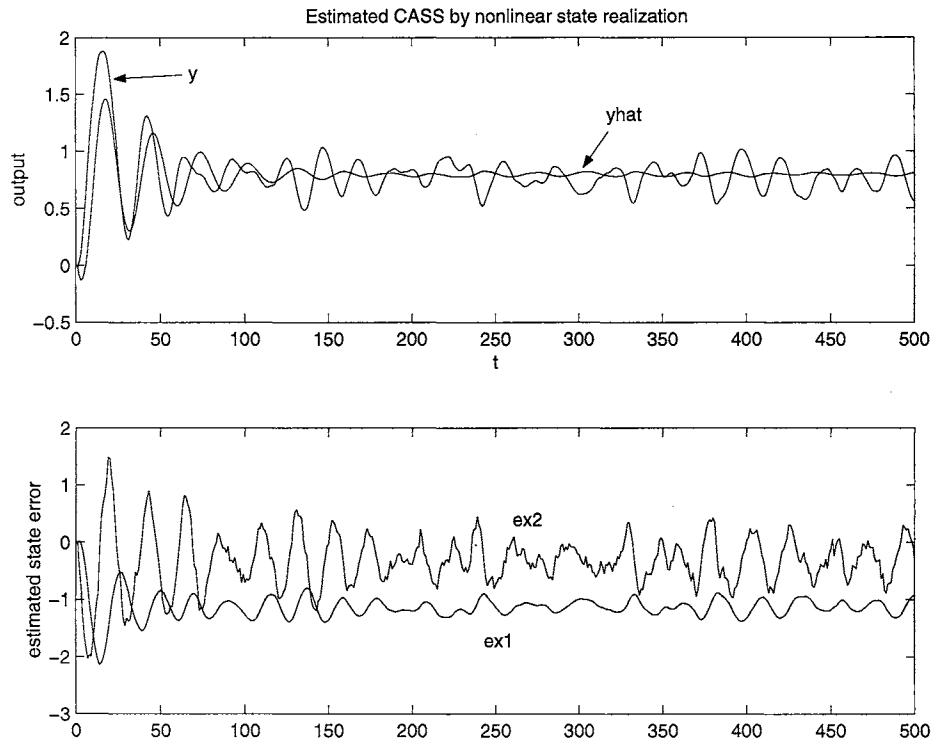


Figure 5.4: Error of CASS model compared with nonlinear system; outputs (Top figure), state estimation error (Bottom figure).

## 5.7 Discussion

In this chapter, we illustrated state realization of nonlinear recursive i/o equations and showed how the state realization can be used for the identification of minimal CASS model. This indirect approach is proposed to overcome the unidentifiability property of CASS model. However, obtaining minimal realization is demanding job if not impossible. We showed that the reduction of observable realization to minimal realization cannot be done in local level but have to be considered using nonlinear isomorphic function.

# Chapter 6

## Conclusion and Future study

### 6.1 Conclusions

This study is about nonlinear system identification for control by multimodels. Our initial work was influenced by literatures in multimodels identification and control, that is, to extend linear technologies to nonlinear systems by heuristics. Later, we turned out interest to multimodels based identification that has more sound theoretical basis. Motivated from multimodels based control, we considered general nonlinear systems represented in state space as the data generating system and try to develop a system identification methodology to recover the data generating system. This approach of attempting to interpret multimodels identification revealed that conventional multimodels identification methods have no clear connection with data generating systems in terms of model structures. In the presence of ambiguity of the conventional multimodels identification, applying model based control to the identified multimodels is bound to be at most heuristic.

We draw the following conclusions:

1. Extension of linear technologies to nonlinear systems is limited as shown in the Laguerre basis function based identification. Even though the proposed algorithm showed impressive response in adaptive filtering to certain type of systems, the general applicability must be questioned, especially for control

related applications.

2. Despite the intuitive appeal of multimodel approaches to system identification and control, it is difficult to make clear connection between i/o based multimodels structure and the general nonlinear systems in the conventional state space
3. Despite the definite need of CASS model identification for control, CASS model is not directly identifiable. Implication of this property for CASS model is quite serious and contradicts the claim of divide-and-conquer.
4. The indirect approach to identification of CASS model is promising. By the indirect approach, we can take advantage of expressive power of the conventional i/o based nonlinear system identification as well as the transparency of multimodels. Also, this approach has clear interpretation of the model structure in connection with conventional system theories based on state space representation.
5. In conclusion, we have to be cautious in implementing multimodels based control with multimodels identification. Model development that has close connection with system theories is important as the reliability of mathematical models is critical in order to secure theoretical reliability of model based control. This is also essential to extend the reliability of control system to uncertain systems, that is, for robust control.

## 6.2 Further works

Since this is the first work ever taken to introduce rigor to multimodel frameworks for system identification in connection with control to the author's knowledge, we face with a wealth of open topics and problems. We provide a list of the open problems

in the following list.

**Rigorous treatment of true systems and the implications in system identification** In this study, we considered the true systems in state space. More rigorous study is necessary to validate this assumption since the scope of nonlinear systems may be too abstract and general. Without the knowledge of the data generating systems, which is typical for system identification problems, the implication of true systems might be quite difficult to be appreciated.

**Nonlinear regression model for data based minimal state realization** Since it is shown that direct identification of local state space model is not possible, the identification of intermediate nonlinear regression model is essential for the estimation of states for CASS model identification. Even though the research on discrete-time state realization has been taken for a while, most of the works are focused on equation based state realization, i.e. transforming regression equation into state space by mathematical manipulations.

Sadegh (2001) is the closest work related to state realization from nonlinear black box system identification. However, the condition to satisfy, for the existence of state realization, is quite difficult to verify and is also difficult to be used for the design of nonlinear regression model. Since the sufficient condition for the existence of state space in (Sadegh, 2001) is only for a special states defined by (5.62), more relaxed conditions for the design of nonlinear regression model structure is possible. For example, inspired by the recursive definition of linear state realization, we can define states recursively if the regression model is a product of functions of i/o's with different lags. That is, if the local nonlinear dynamics can be represented by the model

$$y(t) = \prod_{i=1}^m f_i(y(t-i))g_i(u(t-i)), \quad (6.1)$$

it is straightforward to have state realization as

$$\begin{aligned}
x_1(t) &= y(t) \\
x_1(t+1) &= f_1(x_1(t))g_1(u(t)) \prod_{i=2}^m f_i(y(t-i))g_i(u(t-i)) \\
&= f_1(x_1(t))g_1(u(t))x_2(t) \\
&\vdots \\
x_m(t+1) &= f_m(x_1(t))g_m(u(t)).
\end{aligned}$$

This realization is observable which can be shown by taking Jacobian of collected output vector and the Jacobian is not singular. That is, consider the collected output vector

$$\begin{aligned}
y(t) &= x_1(t), \\
y(t+1) &= x_1(t+1) \\
&= f_1(x_1(t))g_1(u(t))x_2(t), \\
y(t+2) &= f_1(x_1(t+1))g_1(u(t+1))x_2(t+1) \\
&= f_1(f_1(x_1(t))g_1(u(t))x_2(t))g_1(u(t+1))f_2(x_1(t))g_2(u(t))x_3(t), \\
&\vdots \\
y(t+m-1) &= F_{m-1}(x_1(t), \dots, x_m(t), u(t), \dots, u(t+m-1)).
\end{aligned}$$

and the Jacobian matrix is

$$D_x(t)y_{t+m-1}^t = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ \times & f_1(\cdot)g_1(\cdot) & 0 & \dots & 0 \\ \times & \times & f_1(\cdot)g_1(\cdot)f_2(\cdot)g_2(\cdot) & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \end{bmatrix} \quad (6.2)$$

where  $D_x(t)y_{t+m-1}^t$  is the Jacobian matrix, i.e.  $\frac{\partial y_{t+m-1}^t}{\partial x(t)}$ . This is a lower triangular matrix with determinant

$$|D_x(t)y_{t+m-1}^t| = 1 \cdot (f_1(\cdot)g_1(\cdot)) \cdot (f_1(\cdot)f_2(\cdot)g_1(\cdot)g_2(\cdot)) \cdot \dots \quad (6.3)$$

This determinant is not zero unless  $f_i(\cdot)$  or  $g_i(\cdot)$  is independent of  $y(t - i)$  or  $u(t - i)$ , respectively. However, this type of regression model is not conventional in the sense that they are not spanned by basis functions even though this may be more general function class than the one proposed by (Sadegh, 2001). Therefore, we need to establish clear connection between regression models for function approximation or system identification and the regression models for state realization.

**State realization from regression models** As mentioned before, most of works regarding state realization are equation based, utilizing algebraic geometry or differential geometry. Interpreting these works to data based realization methods can open up new horizon of state realization for nonlinear black box models.

**Multi-inputs and multi-outputs** In this work, we only considered SISO case for simplicity and ease of development of ideas. For practical problems, the need to handle MIMO systems is inevitable. Indeed, introduction of more measurements may relax the unidentifiability of CASS model as for the case of biased state estimation (Chmielewski and Klata, 1995).

**Stochastic disturbances** In linear system identification, the effects on parameter estimation and the model error of stochastic disturbances have been the major issues. However, the stochastic effects on model error for general nonlinear systems have rarely attended seriously, probably because the effect of stochastic disturbances on nonlinear dynamical systems is not clearly understood. The introduction of multi-model framework may ease the problem by providing simple local model structures and study the effects of local stochastic disturbances.

**Experiment design for local model identification** The problem of handling global data using multimodels have been recognized in (Shorten et al., 1999). The



localization of data has been handled in heuristic ways as reviewed in Section 1.2 while only localized data is assumed in this work. Isolation of local data for local dynamics is the key issue in multimodel identification, which may also involve closed loop experiment and identification in order to control the experiment.

**Model uncertainty identification** One of the most challenging as well as important issue is to assess the model uncertainty, in order to realize a control system with analytically guaranteed robustness. The indirect identification of local model makes this process more difficult since the model uncertainty of nonlinear models is very difficult to estimate. This issue may be handled by adopting basis function approaches, which is popular for linear system identification, and assess the model uncertainty between nominal truncated model and infinite series model.

**Multimodel based control** The emerging control technologies such as hybrid systems and model based fuzzy control have been experiencing rapid development. Keeping track of the developments in control design and reshaping of the identification area are important in order to avoid the incompatible development of the two disciplines.

# Bibliography

- L. A. Aguirre. A nonlinear dynamical approach to system identification. *Circuits and systems*, 11:10–23,47, 2000.
- P. Andersson. Adaptive forgetting in recursive identification through multiple models. *International Journal of Control*, 42:1175–1193, 1985.
- P. J. Antsaklis. Guest editorial hybrid control systems : An introductory discussion to the special issue. *IEEE Transactions on Automatic Control*, 43:457–459, 1998.
- P. Apkarian and R. J. Adams. Advanced gain-scheduling techniques for uncertain systems. *IEEE Transactions on Control Systems Technology*, 6:21–32, 1998.
- P. Apkarian, P. Gahinet, and G. Becker. Self-scheduled  $H_\infty$  control of linear parameter-varying systems : A design example. *Automatica*, 31:1251–1261, 1995.
- H. H. W. Belt and A. C. D. Brinker. Laguerre filters with adaptive pole optimization. In *Circuits and Systems, 1996. ISCAS '96, Connecting the World, 1996 IEEE International Symposium on*, pages 37–40, 1996.
- C. Bembenek, T. A. Jr. Chmielewski, and P. R. Kalata. Observability conditions for biased linear time invariant systems. In *Proceedings of the American Control Conference, Philadelphia, PA*, pages 1180–1184, 1998.
- W. L. Brogan. *Modern Control Theory*. Prentice-hall, 1991.
- S. G. Cao, N. W. Rees, and G. Feng. Analysis and design of fuzzy control systems using dynamic fuzzy global models. *Fuzzy Sets and Systems*, 75:47–62, 1995.
- S. G. Cao, N. W. Rees, and G. Feng. Analysis and design for a class of complex control systems part II: Fuzzy controller design. *Automatica*, 33:1029–1039, 1997a.
- S. G. Cao, N. W. Rees, and G. Feng. Analysis and design for a class of complex control systems part I: Fuzzy modelling and identification. *Automatica*, 33:1017–1028, 1997b.
- M. J. Chappell and K. R. Godfrey. Global identifiability of the parameters of nonlinear systems with specified inputs: A comparison of methods. *Mathematical Biosciences*, 102:41–73, 1990.

- M.J. Chappell, N.D. Evans, K.R. Godfrey, and M.J. Chapman. Structural identifiability of controlled state space systems: A quasiautomated methodology for generating identifiable reparameterisations of unidentifiable systems. In *Symbolic Computation for Control (Ref. No. 1999/088)*, IEE Colloquium on, pages 8/1 – 8/12, 1999.
- S. Chen, C. F. N. Cowan, and P. M. Grant. Orthogonal least squares learning algorithms for radial basis function networks. *IEEE transactions on neural networks*, 2:302–309, 1991.
- T. A. Chmielewski and P. R. Klata. On the identification of stochastic biases in linear time invariant systems. In *Proceedings of the American Control Conference, Seattle, Washington*, pages 4067–4071, 1995.
- J. Choi, E. A. Misawa, and G. E. Young. Study on sliding mode state estimation. *Journal of dynamic systems, measurement and control*, 121:255–260, 1999.
- J. E. Jr. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, 1983.
- H. Diaz and A. A. Desrochers. Modeling of nonlinear discrete-time systems from input-output data. *Automatica*, 24:629–641, 1988.
- S. V. Drakunov and V. I. Utkin. Sliding mode control in dynamic systems. *International journal of control*, 55:1029–1037, 1992.
- Z. Fejzo and H. Lev-Ari. Adaptive laguerre-lattice filters. *IEEE Transactions on Signal Processing*, 45:3006–3016, 1997.
- D. C. Foley and N. Sadegh. Modelling of nonlinear systems from input-output data for state space realization. In *Proceedings of 40th IEEE Conference on Decision and Control, Orlando, FL*, pages 2980–2985, 2001.
- B. Friedland. Treatment of bias in recursive filtering. *IEEE transactions on automatic control*, AC-14:359–367, 1969.
- Y. Fu and G. A. Dumont. An optimum time scale for discrete Laguerre network. *IEEE transactions on automatic control*, 38:934–938, 1993.
- K. Funahashi and Y. Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks*, 6:801–806, 1993.
- K. Furuta. Sliding mode control of a discrete system. *Systems & control letters*, 14: 145–152, 1990.
- M. Ge, M. Chiu, and Q. Wang. An extended self-organizing map for nonlinear system identification. In *Proceedings of the 38th Conference on Decision and Control, Phoenix, Arizona USA*, pages 1065–1070, 1999.

- M. Green and D. J. N. Limebeer. *Linear Robust Control*. Prentice hall, 1995.
- D. E. Gustafson and W. C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proceedings of IEEE Control and Decision Conference, San Diego, CA*, pages 761–766, 1979.
- R. Haber. Nonlinearity tests for dynamic processes. In *7th IFAC/IFIP Symposium on Identificatin and System Parameter Estimation, York, U.K.*, pages 409–413, 1985.
- R. Haber and H. Unbehauen. Structure identification of nonlinear dynamic systems - a survey on input/output approaches. *Automatica*, 26:651–677, 1990.
- M. T. Hagan, H. B. Demuth, and M. Beale. *Neural Network Design*. PWS publishing company, 1996.
- Richard G. Hakvoort. *System Identification for Robust Process Control*. PhD thesis, Delft university of technology, 1994.
- J. Hammer. Non-linear systems: Stability and rationality. *International Journal of Control*, 40:1–35, 1984.
- Özgüner Ü. Haskara, İ and V. Utkin. On sliding mode observers via equivalent control approach. *International journal of control*, 6:1051–1067, 1998.
- S. Haykin. *Adpative Filter Theory*. Prentice-Hall, 1996.
- K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop. Neural networks for control systems-a survey. *Automatica*, 28:1083–1112, 1992.
- A. Isidori. *Nonlinear Control Systems*. Springer, third edition, 1995.
- J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice-Hall, 1997.
- A. Jeffrey. *Complex Analysis and Applications*. CRC press, 1992.
- P. N. Jin, L. Nikiforuk and M. M. Gupta. Approximation of discrete-time state-space trajectories using dynamic recurrent neural networks. *IEEE transactions on automatic control*, 40:1266–1270, 1995.
- T. A. Johansen. *Operating Regime Based Process Modeling and Identification*. PhD thesis, The Norweigan insitute of technology - university of Trondheim, 1994.
- T. A. Johansen. On the interpretation and identification of dynamic Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 8:297–313, 2000.
- T. A. Johansen and B. A. Foss. Identification of nonlinear system structure and parameters using regime decomposition. *Automatica*, 31:321–326, 1995.

- T.A. Johansen, K.J. Hunt, P.J. Gawthrop, and H. Fritz. Off-equilibrium linearisation and design of gain-scheduled control with application to vehicle speed control. *Control engineering practice*, 6:167–180, 1998.
- R. Johansson, M. Verhaegen, and C. T. Chou. Stochastic theory of continuous-time state-space identification. *IEEE transactions on signal processing*, 47:41–51, 1999.
- M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.
- A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg, and Q. Zhang. Nonlinear black-box models in system identification : mathematical foundations. *Automatica*, 31:1725–1750, 1995.
- J. C. Kalkkuhl and M. R. Katebi. Nonlinear control design using gate function approach. In *IEEE Conference on Control Applications, Glasgow, Scotland*, pages 1195–1200, 1993.
- A. Karimi and I. D. Landau. Robust adaptive control of a flexible transmission system using multiple models. *IEEE transactions on control systems technology*, 8:321–331, 2000.
- C. T. Kelley. *Iterative Methods for Optimization*. SIAM, 1999.
- H. K. Khalil. *Nonlinear Systems*. Prentice hall, 1995.
- K. Kiriakidis. Fuzzy model-based control of complex plants. *IEEE transactions on fuzzy systems*, 6:517–529, 1998.
- K. Kiriakidis, A. Grivas, and A. Tzes. Quadratic stability analysis of the Takagi-Sugeno fuzzy model. *Fuzzy Sets and Systems*, 98:1–14, 1998.
- Ü. Kotta, A.S.I. Zinober, and P. Liu. Transfer equivalence and realization of nonlinear higher order input-output difference equations. *Automatica*, 37:1771–1778, 2001.
- A. Kroll, T. Bernd, and S. Trott. Fuzzy network model-based fuzzy state controller design. *IEEE transactions on fuzzy systems*, 8:632–644, 2000.
- D. C. Lai. Signal processing with orthonormalized exponentials. *Mathematics and Computers in Simulation*, 27:409–420, 1985.
- I. J. Leontaritis and S. A. Billings. Input-output parametric models for non-linear systems, part I: Deterministic non-linear systems, partII: Stochastic non-linear systems. *International Journal of Control*, 41:303–344, 1985.
- P. Liu and C. H. Moog. A local minimal realization algorithm for SISO nonlinear systems. In *Proceedings of the American Control Conference, Baltimore, Maryland*, pages 551–552, 1994.

- L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 2nd edition, 1999.
- L. Ljung and T. Glad. On global identifiability for arbitrary model parametrizations. *Automatica*, 30:265–276, 1994.
- P.M. Mäkilä and J.R. Partington. Robust approximate modelling of stable linear systems. *International journal of control*, 58:665–683, 1993.
- R. Malti, S. B. Ekongolo, and J. Ragot. Dynamic SISO and MISO system approximations based on optimal Laguerre models. *IEEE transactions on automatic control*, 43:1318–1323, 1998.
- D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM journal of applied mathematics*, 11:431–441, 1963.
- M. A. Masnadi-Shirazi and N. Ahmed. Optimum Laguerre networks for a class of discrete-time systems. *IEEE Transactions on Signal Processing*, 39:2104–2108, 1991.
- J. M. Mendel. *Lessons in estimation theory for signal processing, communications, and control*. Prentice Hall, 1995.
- R. Merched and A. H. Sayed. Order-recursive RLS laguerre adaptive filtering. *IEEE Transactions on Signal Processing*, 48:3000–3010, 2000.
- E. A. Misawa. Discrete-time sliding model control: The linear case. *Journal of dynamic systems, measurement and control*, 119:819–821, 1997.
- A. S. Morse, D. Q. Mayne, and G. C. Goodwin. Applications of hysteresis switching in parameter adaptive control. *IEEE transactions on automatic control*, 37:1343–1354, 1992.
- R. Murray-Smith and T. A. Johansen. *Multiple Model Approaches to Modelling and Control*. Taylor & Francis, 1997.
- K. S. Narendra. Neural networks for control: Theory and practice. *Proceedings of the IEEE*, 84:1385–1406, 1996.
- K. S. Narendra and J. Balakrishnan. Adaptive control using multiple models. *IEEE transactions on automatic control*, 42:171–187, 1997.
- K. S. Narendra, J. Balakrishnan, and K. M. Ciliz. Adaptation and learning using multiple models, switching, and tuning. *IEEE Control Systems Magazine*, 15:37–51, 1995.
- K. S. Narendra and S. Mukhopadhyay. Adaptive control using neural networks and approximate models. *IEEE Transactions on Neural Networks*, 8:475–485, 1997.

- K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE transactions on neural networks*, 1:4–27, 1990.
- B. Ninness and F. Gustafsson. A unifying construction of orthonormal bases for system identification. *IEEE transactions on automatic control*, 42:515–521, 1997.
- R.H. Nystöm, K. V. Sandstroöm, T. K. Gustafsson, and H. T. Toivonen. Multimodel robust control of nonlinear plants: A case study. *Journal of process control*, 9: 135–150, 1999.
- K. Ogata. *Discrete-Time Control Systems*. Prentice hall, 1995.
- T. Oliveira e Silva. Optimality conditions for truncated Laguerre networks. *IEEE Transactions on Signal Processing*, 42:2528–2530, 1994.
- T. Oliveira e Silva. On the determination of the optimal pole position of Laguerre filters. *IEEE Transactions on Signal Processing*, 43:2079–2087, 1995.
- S. W. Piché. Steepest descent algorithms for neural network controllers and filters. *IEEE transactions on neural networks*, 2:198–212, 1994.
- A. Pinkus. *N-Widths in Approximation Theory*. Springer-Verlag, 1985.
- J. C. Principe, L. Wang, and M. A. Motter. Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control. *Proceedings of the IEEE*, 86:2240–2258, 1998.
- I. Rojas, H. Pomares, J. Ortega, and A. Prieto. Self-organized fuzzy system generation from training examples. *IEEE transactions on fuzzy systems*, 8:23–36, 2000.
- A. M. Sabatini. A hybrid genetic algorithm for estimating the optimal time scale of linear systems approximations using laguerre models. *IEEE transactions on automatic control*, 45:1007–1011, 2000.
- N. Sadegh. State realization of nonlinear MIMO discrete-time systems described by input-output equations. In *Proceedings of the 37th IEEE Conference on Decision & Control*, 4614–4619, 1998.
- N. Sadegh. Minimal realization of nonlinear systems described by input-output difference equations. *IEEE transactions on automatic control*, 46:698–710, 2001.
- R. M. Sanner and J. J. E. Slotine. Gaussian networks for direct adaptive control. *IEEE transactions on neural networks*, 3:837–863, 1992.
- D. Sbarbaro and T.A. Johansen. Multiple local Laguerre models for modelling nonlinear dynamic systems of the Wiener class. *IEE Proceedings in Control theory and applications*, 144:375–380, 1997.
- M. Setnes. Supervised fuzzy clustering for rule extraction. *IEEE transactions on fuzzy systems*, 8:416–424, 2000.

- J. S. Shamma and M. Athans. Analysis of gain scheduled control for nonlinear plants. *IEEE transactions on automatic control*, 35:898–907, 1990.
- R. Shorten, R. Murray-Smith, R. BJørgan, and H. Gollee. On the interpretation of local models in blended multiple model structures. *International Journal of Control*, 72:620–628, 1999.
- J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification : a unified overview. *Automatica*, 31:1691–1724, 1995.
- A. Skeppstedt, L. Ljung, and M. Millnert. Construction of composite models from observed data. *International journal of control*, 55:141–152, 1992.
- J. E. Slotine. International journal of control. *International journal of control*, 40:421–434, 1984.
- J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice hall, 1991.
- E. Sontag. *Polynomial Response Maps*. New York: Springer-verlag, 1979.
- B. Srinivasan, U. R. Prasad, and N. J. Rao. Back propagation through adjoints for the identification of nonlinear dynamic systems using recurrent neural models. *IEEE transactions on neural networks*, 5:213–228, 1994.
- J. Stark. Adaptive model selection using orthogonal least squares methods. *Proceedings of royal society of london series A - mathematical physical and engineering sciences*, 453:21–42, 1997.
- T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:116–132, 1985.
- K. Tanaka, T. Ikeda, and H. O. Wang. Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: Quadratic stabilizability,  $h^\infty$  control theory, and linear matrix inequalities. *IEEE transactions on fuzzy systems*, 4:1–13, 1996.
- K. Tanaka, T. Ikeda, and H. O. Wang. Fuzzy regulators and fuzzy observers: Relaxed stability conditions and LMI-based designs. *IEEE transactions on fuzzy systems*, 6:250–265, 1998.
- K. Tanaka and M. Sano. A robust stabilization problem of fuzzy control systems and its applications to backing up control of a truck-trailer. *IEEE transactions on fuzzy systems*, 2:119–134, 1994.
- S. Tøffner-Clausen. *System identification and robust control, a case study approach*. Springer, 1996.
- H. Unbehauen. Some new trends in identification and modeling of nonlinear dynamical systems. *Applied mathematics and computation*, 78:279–297, 1996.



- S. Vajda, K. R. Godfrey, and H. Rabitz. Similarity transformation approach to identifiability analysis of nonlinear compartmental models. *Mathematical biosciences*, 93:217–248, 1989.
- J. M. Van Den Hof. Structural identifiability of linear compartmental systems. *IEEE transactions on automatic control*, 43:800–818, 1998.
- B. Wahlberg. System identification using Laguerre models. *IEEE transactions on automatic control*, 36:551–562, 1991.
- B. Wahlberg. Laguerre and Kautz models. In *IFAC Symposium on System Identification, Copenhagen, Denmark*, pages 965–976, 1994.
- B. Wahlberg. Orthonormal basis functions models: A transformation analysis. In *14th Triennial World Congress of IFAC, Beijing, P.R. China*, pages 355–360, 1999.
- E. Walter and L. Pronzato. Qualitative and quantitative experiment design for phenomenological models- a survey. *Automatica*, 26:195–213, 1990.
- L. Wang and W. R. Cluett. Optimal choice of time-scaling factor for linear system approximations using Laguerre models. *IEEE Transactions on Automatic Control*, 39:1463–1467, 1994.
- U. Witkowski, S. Rüping, U. Rückert, F. Schütte, S. Beineke, and H. Grotstollen. System identification using selforganizing feature maps. In *Artificial Neural Networks Conference Publication No. 440*, pages 100–105, 1997.
- J. Yen, L. Wang, and C. W. Gillespie. Improving the interpretability of TSK fuzzy models by combining global learning and local learning. *IEEE transactions on fuzzy systems*, 6:530–537, 1998.
- K. D. Young, V. I. Utkin, and Ü Özgüner. A control engineer’s guide to sliding mode control. *IEEE transactions on control systems technology*, 7:328–342, 1999.
- L. Zhang and W. Z. Rymer. Simultaneous and nonlinear identification of mechanical and reflex properties of human elbow joint muscles. *IEEE transactions on biomedical engineering*, 44:1192–1209, 1997.
- K. Zhou and J. C. Doyle. *Essentials of Robust Control*. Prentice hall, 1998.

# Appendix A

## Overview of unconstrained nonlinear least squares

We give a short overview of unconstrained nonlinear least squares in this section. The presentation is mainly based on (Dennis and Schnabel, 1983; Kelley, 1999).

In contrast to linear Least Squares (LS) problems, nonlinear LS problems generally have no closed form solutions and have to rely on iterative search methods. The LS error criterion to minimize is given as

$$V(\theta) = \frac{1}{2N} E(\theta)^T E(\theta), \quad (\text{A.1})$$

The Newton method is derived from the second order Taylor series approximation of LS measure as

$$V_q(\theta_k + \Delta\theta_k) = V(\theta_k) + \nabla V(\theta_k)^T \Delta\theta_k + \frac{1}{2} \Delta\theta_k^T \nabla^2 V(\theta_k) \Delta\theta_k, \quad (\text{A.2})$$

and is described by

$$\nabla^2 V(\theta_k) \Delta\theta_k = -\nabla V(\theta_k) \quad (\text{A.3})$$

$$\theta_{k+1} = \theta_k + \Delta\theta_k. \quad (\text{A.4})$$

Even though the Newton method has a q-quadratic convergence when it converges, this method requires the second partial derivative  $\nabla^2 V(\theta_k)$  at each iteration and is sensitive to the initial parameter  $\theta_0$  for the convergence. The steepest descent method

is derived from the first order Taylor series and is described as

$$\theta_{k+1} = \theta_k - \lambda_k \nabla V(\theta_k), \quad (\text{A.5})$$

where  $\lambda_k$  is selected to guarantee the decrease in the cost function with newly updated parameter or more stringently to satisfy

$$V(\theta_{k+1}) < V(\theta_k) - \lambda_k \alpha \nabla V(\theta_k)^T \nabla V(\theta_k)$$

with  $\alpha \in (0, 1)$  to obtain a sufficient decrease. The steepest descent method decreases the error at every iteration and eventually achieves a global convergence, however, is quite slow especially when the error surface is poorly conditioned.

One nice feature of LS problems is that partial information of  $\nabla^2 V(\theta_k)$  is available without taking second partial derivatives. Consider the error measure by (A.1), the first derivative, so called gradient, and the second derivatives with respect to  $\theta$ , the so called Hessian, is

$$\frac{\partial V}{\partial \theta} = -\frac{1}{N} \sum_{t=0}^{N-1} \epsilon(t|\theta) \frac{\partial \hat{y}(t|\theta)}{\partial \theta}, \quad (\text{A.6})$$

$$= \frac{1}{N} J(\theta)^T E(\theta)$$

$$\frac{\partial^2 V}{\partial \theta^2} = \frac{1}{N} \sum_{t=0}^{N-1} \left( \frac{\partial \hat{y}(t|\theta)}{\partial \theta} \frac{\partial \hat{y}(t|\theta)^T}{\partial \theta} - \epsilon(t|\theta) \frac{\partial^2 \hat{y}(t|\theta)}{\partial \theta^2} \right). \quad (\text{A.7})$$

$$= \frac{1}{N} (J(\theta)^T J(\theta) + S(\theta)),$$

where  $J(\theta) = \partial E(\theta)/\partial \theta$  and  $S(\theta) = \sum_{t=0}^{N-1} \epsilon(t|\theta) \frac{\partial^2 \epsilon(t)}{\partial \theta^2}$ . In (A.7), the first term of r.h.s is accessible without computing second partial derivatives. By omitting  $S(\theta)$ , the Gauss-Newton method is described as

$$J(\theta_k)^T J(\theta_k) \Delta \theta_k = -J(\theta_k)^T E(\theta_k). \quad (\text{A.8})$$

Another way to the Gauss-Newton method is to consider the first order approximation of  $E(\theta)$  as

$$E(\theta) = E(\theta_k) + J(\theta_k) \Delta \theta_k, \quad (\text{A.9})$$

which can be viewed as the linear LS problem by letting  $E(\theta) = 0$ . Since (A.9) is the Taylor series approximation, the validity of (A.9) depends on the radius of  $\Delta\theta_k$ . The Levenberg-Marquardt algorithm shares the properties of the global convergence of the steepest method and the rapid convergence of Gauss-Newton method. The algorithm is described as

$$(J(\theta_k)^T J(\theta_k) + \mu_k I) \Delta\theta_k = -J(\theta_k)^T E(\theta_k). \quad (\text{A.10})$$

With a small value of  $\mu_k$ , the algorithm behaves like Gauss-Newton, while the algorithm converges to the steepest descent with a small step as  $\mu_k$  becomes large. Even though small  $\mu_k$  is necessary for rapid convergence, large  $\mu_k$  should be used to guarantee the decrease in the cost function when (A.9) is not a good approximation.

There are several methods to adjust  $\mu_k$  in (A.10). In (Marquardt, 1963), adjusting  $\mu_k$  by multiplying or dividing it by an arbitrary factor is proposed while guaranteeing the decrease of the cost function at each iteration. A more sophisticated method would be adjusting the radius of  $\Delta\theta_k$  and solving for  $\mu_k$ , such that  $\Delta\theta_k$  is within the bounded region, which is called the trust region method (Dennis and Schnabel, 1983).

Trust region methods can be applied to the Newton method if the full Hessian is available with reasonable costs. The advantage of the trust region method combined with Newton method is in a smooth transition between the steepest descent and the Newton method to achieve fast global convergence.

# Appendix B

## Discrete time sliding mode controller

In this section, we consider discrete sliding mode control for the explicit compensation of the unknown disturbances but with known bounds. We give constructive derivation of discrete sliding controllers in the presence of unknown but bounded uncertainty. Even sliding mode control technology cannot be directly applied to sliding mode observer, they have close connections. Therefore, this section can be considered as a preliminary step for the derivation of sliding mode observer.

Sliding mode control algorithms are now widely used due to their robustness against bounded uncertainties as well as the decomposition of high dimension problem into a set of independent subproblems of lower dimension (Drakunov and Utkin, 1992). The robustness against bounded uncertainties is the major motivation of considering sliding mode because observer error dynamics has the similarity with the control problem with bounded uncertainty. Sliding mode control has initially developed for continuous time systems (Slotine, 1984), while discrete time sliding mode controller has been developed as well (Furuta, 1990; Misawa, 1997). In this section, we give constructive proofs of stable discrete time controllers and compare them. Compared with sliding mode controllers proposed in (Furuta, 1990; Misawa, 1997), our controller is derived in a straightforward way, illustrating the formation of boundary layer from uncertainty of the model. Also, the second controller does not require discontinuous

action to guarantee the attraction to the boundary layer even in the presence of model uncertainty.

Consider a linear discrete time system given as

$$x(t+1) = Ax(t) + Bu(t) + F \quad (\text{B.1})$$

where  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}$ , and  $F \in \mathbb{R}^n$ .  $F$  is a unknown disturbance vector but with known bound, which will be clarified shortly. We present two different methods to derive discrete sliding mode control. The first one is given in the following theorem.

**Theorem B.1 (Discrete sliding mode controller 1)** Let the sliding mode be defined as

$$s(t) = G\tilde{x}(t) \quad (\text{B.2})$$

where  $\tilde{x}(t) = x(t) - x_d(t)$  with  $x_d(t)$ , the desired states and  $x(t)$  is the state of (B.1), and boundary layer  $\phi = \bar{F}_G + \epsilon_\phi$  with some small positive number  $\epsilon_\phi$ . Then a sliding mode controller to track the desired state within the boundary layer, that is,  $|G\tilde{x}(t)| < \phi$  is given as

$$u(t) = u_{eq}(t) - \Delta u(t) \quad (\text{B.3})$$

where  $u_{eq}(t)$  is defined as

$$u_{eq}(t) = (GB)^{-1}[s(t) - G(Ax(t) + \hat{F} - x_d(t+1))] \quad (\text{B.4})$$

and  $\Delta u(t)$  is the discontinuous controller that is defined as

$$\Delta u(t) = \begin{cases} K \text{sign}(s(t)) & \text{if } |s(t)| > \phi \\ (1+a)s(t) & \text{if } |s(t)| \leq \phi \end{cases} \quad (\text{B.5})$$

Here,  $K$  is a positive constant satisfying

$$\bar{F}_G < K < 2|s(t)| - \bar{F}_G \quad (\text{B.6})$$

and  $a$  is a constant satisfying

$$0 < a < 1 - \frac{\bar{F}_G}{\phi} = \frac{\epsilon_\phi}{\phi}. \quad (\text{B.7})$$

Here  $\bar{F}_G$  is the uncertainty bound satisfying

$$|G\tilde{F}| < \bar{F}_G \quad (\text{B.8})$$

where  $\tilde{F} = F - \hat{F}$ .

**Proof:** Define a sliding surface as

$$s(t) = G\tilde{x}(t) \quad (\text{B.9})$$

where  $\tilde{x}(t) = x(t) - x_d(t)$  with  $x_d(t)$ , the desired states and  $x(t)$  is the state of (B.1).  $G$  is a constant row vector and is selected so that it satisfies stability and performance specifications.

Firstly, we can obtain an equivalent controller  $u_{eq}(t)$  from nominal model and a condition of sliding mode  $s(t+1) - s(t) = 0$  (Young et al., 1999) as,

$$s(t+1) - s(t) = G\tilde{x}(t+1) - s(t) \quad (\text{B.10})$$

$$= G[Ax(t) + Bu(t) + \hat{F} - x_d(t+1)] - s(t) = 0 \quad (\text{B.11})$$

Solve for  $u(t)$  and we obtain

$$u_{eq}(t) = (GB)^{-1}[s(t) - G(Ax(t) + \hat{F} - x_d(t+1))] \quad (\text{B.12})$$

where  $\hat{F}$  is a nominal of  $F$ .

Now, for the convergence to sliding mode, sliding condition must be satisfied. The sliding condition can be obtained from discrete Lyapunov function such as

$$V = s(t)^2. \quad (\text{B.13})$$

For asymptotic stability,

$$V(t+1) < V(t) \Rightarrow$$

$$s(t+1)^2 - s(t)^2 = (s(t+1) - s(t))(s(t+1) + s(t)) \quad (\text{B.14})$$

$$= \Delta s(t)(\Delta s(t) + 2s(t)) < 0. \quad (\text{B.15})$$

where  $\Delta s(t) = s(t+1) - s(t)$ .  $\Delta s(t)$  with uncertainty results in

$$\Delta s(t) = G(F - \hat{F}) - \Delta u(t) \quad (\text{B.16})$$

$$= G\tilde{F} - \Delta u(t) \quad (\text{B.17})$$

when  $u(t) = u_{eq}(t) - (GB)^{-1}\Delta u(t)$  is applied, where  $\Delta u(t)$  is extra freedom to compensate the uncertainty. Hence, the sliding condition becomes

$$\Delta s(t)(\Delta s(t) + 2s(t)) = (G\tilde{F} - \Delta u(t))(G\tilde{F} - \Delta u(t) + 2s(t)) < 0 \quad (\text{B.18})$$

Solve the inequality and we obtain

$$G\tilde{F} < \Delta u(t) < G\tilde{F} + 2s(t), \quad \text{if } s(t) > 0 \quad (\text{B.19})$$

$$G\tilde{F} + 2s(t) < \Delta u(t) < G\tilde{F}, \quad \text{if } s(t) < 0 \quad (\text{B.20})$$

Suppose that

$$-\bar{F}_G < G\tilde{F} < \bar{F}_G \quad (\text{B.21})$$

and (B.19) and (B.20) are satisfied if

$$\bar{F}_G < \Delta u(t) < 2s(t) - \bar{F}_G \quad \text{if } s(t) > 0 \quad (\text{B.22})$$

$$2s(t) + \bar{F}_G < \Delta u(t) < -\bar{F}_G \quad \text{if } s(t) < 0 \quad (\text{B.23})$$

which can be simplified as

$$\bar{F}_G < \Delta u(t) < 2s(t) - \bar{F}_G \quad (\text{B.24})$$

$$\bar{F}_G < -\Delta u(t) < -2s(t) - \bar{F}_G = 2|s(t)| - \bar{F}_G \quad (\text{B.25})$$

This is satisfied if

$$\Delta u(t) = K \text{sgn}(s(t)) \quad (\text{B.26})$$

where  $\bar{F}_G < K < 2|s(t)| - \bar{F}_G$ . The inequalities make sense if

$$\bar{F}_G < 2|s(t)| - \bar{F}_G, \quad (\text{B.27})$$



that is,

$$\bar{F}_G < |s(t)|. \quad (\text{B.28})$$

Therefore, the boundary layer of

$$S = \{\tilde{x}(t) : |s(t)| \leq \bar{F}_G\} \quad (\text{B.29})$$

is attractive and we may consider this as the boundary layer for our sliding mode controller.

Now, check the invariance of the boundary layer, that is, if the states inside the boundary layer remains inside it. From (B.17), we obtain

$$s(t+1) = s(t) - \Delta u(t) + G\tilde{F} \quad (\text{B.30})$$

If we choose  $\Delta u(t)$  simply as  $\Delta u(t) = s(t)$ , then

$$s(t+1) = G\tilde{F} < \bar{F}_G \quad (\text{B.31})$$

which shows that the boundary layer is invariant by the assumption of (B.21). In general, we want asymptotic stability or at least to minimize the effect of disturbances. Hence, it is more attractive to choose  $\Delta u(t) = (1+a)s(t)$  and we obtain

$$s(t+1) = -as(t) + G\tilde{F}. \quad (\text{B.32})$$

We want  $-1 < a < 1$  for the stability of (B.32) and  $|1+a| > 1$  to reduce the effect of uncertainty since

$$s(\infty) = \frac{G\tilde{F}}{1+a} \quad (\text{B.33})$$

in steady state. Hence, the desired  $a$  is

$$0 < a < 1. \quad (\text{B.34})$$

Now, check the invariance of boundary layer with this controller. Suppose that the sliding surface is inside the boundary layer, that is,

$$-\bar{F}_G \leq s(t) \leq \bar{F}_G. \quad (\text{B.35})$$

From (B.35), (B.34), (B.32) and (B.21), we obtain

$$-2\bar{F}_G < s(t+1) = (-as(t) + G\tilde{F}) < 2\bar{F}_G \quad (\text{B.36})$$

which clearly shows that the boundary layer  $|s(t)| \leq \bar{F}_G$  is not invariant. In order to satisfy the invariance property, we expand the interval of boundary layer. Denote the enlarged boundary as  $\phi$  and suppose that  $s(t) \leq \phi$ , then we obtain

$$-a\phi - \bar{F}_G < s(t+1) = (-as(t) + G\tilde{F}) < a\phi + \bar{F}_G \quad (\text{B.37})$$

from (B.32), (B.34) and (B.21). Since we want

$$-\phi \leq s(t+1) \leq \phi, \quad (\text{B.38})$$

we can easily see that

$$a\phi + \bar{F}_G < \phi \quad (\text{B.39})$$

Let  $\phi = \bar{F}_G + \epsilon_\phi$  and we can obtain from (B.39) and (B.34),

$$0 < a < 1 - \frac{\bar{F}_G}{\phi} = \frac{\epsilon_\phi}{\phi}. \quad (\text{B.40})$$

Now, we can conclude that the control input  $\Delta u(t) = (1+a)s(t)$  with  $a$  satisfying (B.40) make the boundary layer  $\phi = \bar{F}_G + \epsilon_\phi$  invariant with some positive number  $\epsilon_\phi$ .

□

We can derive different sliding mode controller based on a different condition of sliding mode. We present the second sliding mode controller in the following theorem.

**Theorem B.2 (Discrete sliding mode controller 2)** Let the sliding mode be defined as

$$s(t) = G\tilde{x}(t) \quad (\text{B.41})$$

where  $\tilde{x}(t) = x(t) - x_d(t)$ , same as Theorem B.1. Then a sliding controller that track the desired states within boundary layer is given as

$$u(t) = u_{eq}(t) + \Delta u(t) \quad (\text{B.42})$$

where

$$u_{eq}(t) = -(GB)^{-1}G\{Ax(t) + \hat{F} - x_d(t+1)\} \quad (\text{B.43})$$

and

$$\Delta u(t) = \begin{cases} 0 & \text{if } |s(t)| > \phi \\ as(t) & \text{if } |s(t)| \leq \phi \end{cases} \quad (\text{B.44})$$

Here  $a$  must satisfy

$$0 < a < 1 - \frac{\bar{F}_G}{\phi} = \frac{\epsilon_\phi}{\phi} \quad (\text{B.45})$$

where the bound of boundary layer is defined as  $\phi = \bar{F}_G + \epsilon_\phi$  with some small positive number  $\epsilon_\phi$ .

**Proof:** Same as Theorem B.1, define a sliding surface as

$$s(t) = G\tilde{x}(t) \quad (\text{B.46})$$

Equivalent control can be obtained with nominal model but with different condition of sliding mode as

$$s(t+1) = G(Ax(t) + Bu(t) + \hat{F} - x_d(t+1)) = 0 \quad (\text{B.47})$$

Solve for  $u(t)$  and we obtain

$$u_{eq}(t) = -(GB)^{-1}G\{Ax(t) + \hat{F} - x_d(t+1)\} \quad (\text{B.48})$$

By adding extra freedom to compensate the uncertainty as  $u(t) = u_{eq}(t) - (GB)^{-1}\Delta u(t)$  and we obtain

$$s(t+1) = G\tilde{x}(t+1) = G(F - \hat{F}) - \Delta u(t) = G\tilde{F} - \Delta u(t) \quad (\text{B.49})$$

The sliding condition given from Theorem B.1 is represented with the above equation as

$$(s(t+1) - s(t))(s(t+1) + s(t)) = (G\tilde{F} - \Delta u(t) - s(t))(G\tilde{F} - \Delta u(t) + s(t)) < 0 \quad (\text{B.50})$$

The above inequality is satisfied if and only if

$$G\tilde{F} - s(t) < \Delta u(t) < G\tilde{F} + s(t) \quad \text{if } s(t) > 0 \quad (\text{B.51})$$

$$G\tilde{F} + s(t) < \Delta u(t) < G\tilde{F} - s(t) \quad \text{if } s(t) < 0 \quad (\text{B.52})$$

Suppose  $|s(t)| > G\tilde{F}$  and we obtain

$$\bar{F}_G - s(t) < \Delta u(t) < s(t) - \bar{F}_G \quad \text{if } s(t) > 0 \quad (\text{B.53})$$

$$\bar{F}_G + s(t) < \Delta u(t) < -s(t) - \bar{F}_G \quad \text{if } s(t) < 0 \quad (\text{B.54})$$

which is reduced to one equation as

$$-(|s(t)| - \bar{F}_G) < \Delta u(t) < |s(t)| - \bar{F}_G. \quad (\text{B.55})$$

where the inequality makes sense if  $|s(t)| > \bar{F}_G$ . From (B.55), we can see that  $\Delta u(t)$  is not necessary even in the presence of model uncertainty, in order to attract the tracking error to the boundary layer  $|s(t)| > \bar{F}_G$ .

Now, check the invariance of the boundary layer. If we select  $\Delta u(t) = 0$  inside the boundary layer, i.e.  $s(t) < \bar{F}_G$ , then

$$s(t+1) = G\tilde{F} < \bar{F}_G \quad (\text{B.56})$$

which shows the invariance. If we are interested in reducing the effect of uncertainty, we can choose nonzero  $\Delta u(t)$  inside the boundary layer, similar with Theorem B.1 as

$$\Delta u(t) = as(t) \quad (\text{B.57})$$

and the sliding dynamics becomes

$$s(t+1) = -as(t) + G\tilde{F} \quad (\text{B.58})$$

For stability of the sliding surface,

$$-1 < a < 1 \quad (\text{B.59})$$

Similarly with Theorem B.1,  $|s(t)| \leq \bar{F}_G$  is not invariant with this choice of  $\Delta u(t)$ . Choose larger boundary layer as  $\phi = \bar{F}_G + \epsilon_\phi$  with some positive number  $\epsilon_\phi$  and we obtain

$$0 < a < 1 - \frac{\bar{F}_G}{\phi} \tag{B.60}$$

by similar analysis with Theorem B.1. □

**Remark B.1** *We can see that the second sliding mode controller is resulted in much simpler controller without any discontinuity, which is a huge advantage over discontinuous control.*

δ  
VITA

Seok-beom Lee

Candidate for the Degree of

Doctor of Philosophy

Thesis: BLACK BOX SYSTEM IDENTIFICATION OF A NONLINEAR SYSTEM  
FOR THE DESIGN OF MODEL BASED CONTROL BY MULTIMODELS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Seoul, the Republic of Korea, On April 11, 1973, the son of Lee, Kyehong and Sin, Jeungrye

Education: Graduated from Kon-kuk University, Seoul, Korea with the Degree of Bachelor of Engineering in Mechanical Design and Production Engineering in February 1995; received the Degree of Master of Science in Mechanical Engineering, Oklahoma State University, in December, 1997. Completed the requirements for the Degree of Doctor of Philosophy with a major in Electrical Engineering at Oklahoma State University in May, 2002.

Experience: Teaching and Research Assistant of Electrical Engineering Department, Oklahoma State University (1/1998-12/2001).